A GUIDED TOUR OF PERSONAL COMPUTING

By the Sorcerer of Exidy inc.

A GUIDED TOUR
OF
PERSONAL COMPUTING

By
The Sorcerer of Exidy

390 Java Drive
Sunnyvale, California 94086

SECOND EDITION
FEBRUARY 1979

PRINTED IN U.S.A.

TRAVEL ITINERARY

APPENDIXES

WELCOME ABOARD

This book is an all inclusive tour of the wonderful world of personal computing, with your very own Sorcerer as a tour guide. Here's a quick idea of where we are going, how we plan to get there, and some of the more interesting sights along the way.

PACKING AND TRAVEL AIDS start you out on the right foot as a first time traveler. You will discover how to handle your Sorcerer and should problems occur, what to do about them. We will explore the concept of a computer system and what is involved in putting one together. Finally you will want to talk to your Sorcerer, so we will explain how that is done.

BON VOYAGE AND INDIVIDUAL TOURING are for everyone. You will discover that your conversation with Sorcerer will be conducted by typing and reading messages displayed on a television-like screen. A detailed trip through the keybord will make you aware of the power and flexibility of your Sorcerer. Individual tour packages are introduced as useful extensions of your Sorcerer's capability.

THE EXPERIENCED TRAVELER will not want to miss Scheduled Program, Special Offering and Further Information. We will explain the language of Sorcerer's Power-On Monitor, and teach you how to create your very own symbols or character set to be displayed on the screen. You will find the hidden capability of a communications interface and printer connection for that information you need written down. A detailed description of the essential add-on components (video monitor and cassette tape drive) gives you a complete understanding of their capability and use.

THE APPENDIX has a wealth of information for those who like detail. A memory map of program locations shows which memory spaces are available for your use, and which ones the Sorcerer has reserved for itself. The Z80 microprocessor instructions are listed by both mnemonic and op-code. For those who want to explore far away places a list of hardware options if provided, and there is a glossary of buzz words.

As you read this manual you will discover that learning can be fun. The Sorcerer will introduce you to general purpose digital computing and provide an instant response as you experiment with its operation and programming. There's no waiting and wondering if you performed an operation correctly, Sorcerer will let you know who's in command. Let's get started!

PACKING AND UNPACKING

(System Hook-Up and Turn-On)

Please check the package for damage. If there is any, report this IMMEDIATELY to the shipping agent. Examine your Sorcerer carefully for concealed damage and report this IMMEDIATELY to an authorized Exidy dealer, or to Exidy Customer Service.

CUSTOMER SERVICE INFORMATION

All inquiries relating to service or repair of Exidy computer products should be directed to an authorized Exidy dealer or to Customer Service, Data Products Division, 390 Java Drive, Sunnyvale, California 94086. You must have a return authorization number for all merchandise returned.

Carefully unpack the Sorcerer. Remove all packing material. Be sure to locate all cables, documentation, cartridges, etc. Save the shipping carton in case you need to transport the Sorcerer.

CHECKLIST: Be sure you have each item.

- ☐ Sorcerer Computer, 1 each
- ☐ Video Display Cable, 1 each
- ☐ Cassette Recorder Cable, 2 each
- ☐ Guided Tour Manual, 1 each

Place the Sorcerer on a counter, desk or other suitable surface, making sure you don't block air passage to the ventilation holes in the Sorcerer case. Plug Sorcerer into any standard, grounded electrical outlet with 120 volt, 60Hz AC power. (Note: Sorcerer can be factory set for operation of 220 volt, 50Hz AC power.)

INSTALLATION TIPS AND SAFETY POINTS

- *WHERE TO INSTALL YOUR SORCERER:*

  Choose a place where sunlight does not shine directly on the screen.

  The set has air vents at the bottom and at the back to prevent overheating. These vents should be uncovered at all times to allow for proper ventilation.

  Avoid dusty places, and do not place your Sorcerer near radiators or other heat sources. Overheating can lead to faulty performance.

  Electrical appliances should always be kept away from moisture, and your Sorcerer is no exception!

- *DO NOT ALLOW ANYTHING TO BE PUSHED INSIDE THE COMPUTER THROUGH THE AIR VENTS.*

- *NEVER REMOVE THE TOP COVER. REMOVAL OF THE COVER PROVIDES ACCESS TO THE ELECTRICAL COMPONENTS AND UNQUALIFIED PERSONS TOUCHING THESE MAY RECEIVE BURNS AND/OR ELECTRICAL SHOCKS.*

- *DO NOT SUBJECT YOUR SORCERER TO IMPACT OF ANY KIND.*

- *DURING A THUNDERSTORM IT IS ADVISABLE TO UNPLUG YOUR SORCERER FROM THE MAIN ELECTRICAL POWER.*

- *IF THE COMPUTER PRODUCES ABNORMAL SOUND OR SMELL DURING ITS OPERATION, TURN THE POWER OFF, UNPLUG THE UNIT AND CONTACT YOUR DEALER.*

- *DO NOT USE ALCOHOL, BENZENE, THINNER OR CHEMICAL SOLVENT TO CLEAN THE CASE. A SOFT CLOTH DAMPENED WITH WATER IS ALL YOU NEED.*

INSERTING A ROM PAC™

ROM PACs are plug-in program cartridges which resemble 8-track stereo cartridges. They plug directly into the side of Sorcerer. When properly inserted, a ROM PAC automatically becomes part of Sorcerer's memory (see Appendix A). Your Exidy dealer has ROM

PACs containing languages and operating systems such as Standard BASIC, a Word Processor, and a Development PAC™ (Assembler/Editor). Insert these cartridges into the side of the Sorcerer (label side up) when the AC power is OFF.

CAUTION: Cartridges should be inserted or removed from Sorcerer ONLY when the AC power is off, to protect the programs.

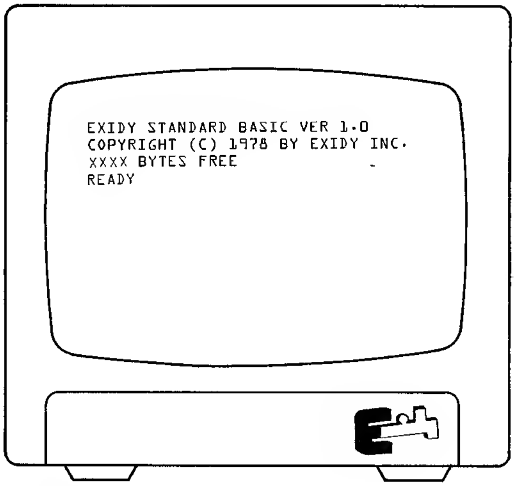
Turn the power to Sorcerer off by depressing the rocker switch at the back of the unit. Insert the ROM PAC into the side of Sorcerer (label side up). Push on the cartridge until it feels like it is in place with its mating connector. The cartridge indentations for gripping should be exposed outside Sorcerer.

CONNECTING THE VIDEO DISPLAY

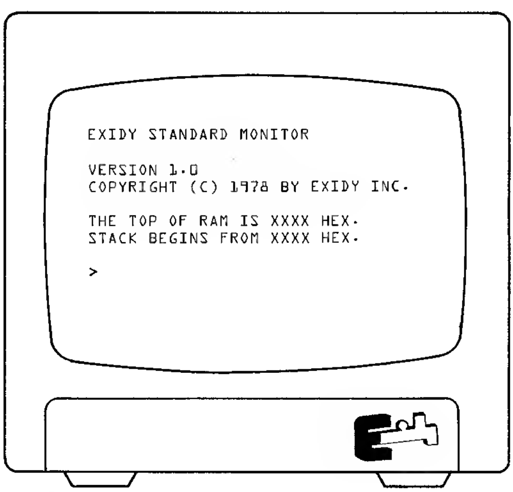
The video signal output at the rear video connector of Sorcerer is composite video (refer to Figure 1). The cable provided has a mating connector for both the Sorcerer output and the Exidy video display. (Note: An approved second source of video display for the Sorcerer is the Hitachi P/05M receiver monitor.)

If you do not have a video display monitor you can use a standard black and white television *if* you have an FCC approved RF modulator. The modulator connects the video output of Sorcerer directly to the antenna lead-in connection of your television set. This is cumbersome and produces a more distorted picture than a direct video display monitor.

When the video cable is attached to the Sorcerer and power is applied to both the display and the Sorcerer, a message will appear on the screen:



With a ROM PAC inserted correctly into the side of Sorcerer, you will see a ROM PAC sign-on message on the screen. This is the Standard BASIC signature.



Without a ROM PAC inserted into the side of Sorcerer, the internal Power-On Monitor program is automatically executed and the above message displayed.

If you do not see a message on the screen, proceed as follows:

1. Check the AC power outlet to the Sorcerer and video monitor, by plugging in a lamp and making sure it comes on.

2. Turn the power switch off and then on again at the back of the Sorcerer.

3. Turn up the brightness control on the monitor until the screen is almost white. (Note: The brightness and contrast controls should be set to a comfortable reading intensity. ***CAUTION:*** Setting the intensity too high may damage the screen.)

4. Check the connectors on both ends of the video cable to make sure they are secure.

IF YOU STILL DO NOT SEE THE APPROPRIATE MESSAGE ON THE SCREEN, CONTACT YOUR EXIDY DEALER FOR HELP.

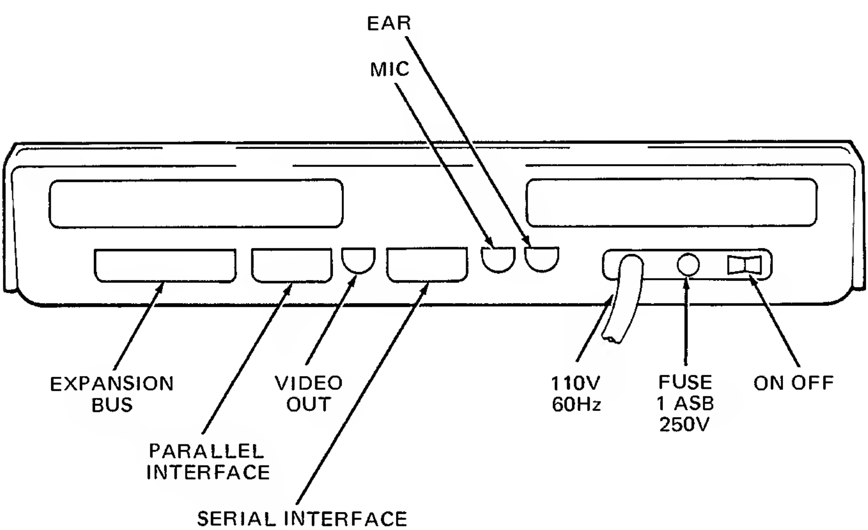


Figure 1. Connector Locations at the Back of Sorcerer

CONNECTING THE CASSETTE RECORDER

It is not necessary to connect the cassette recorder unless you plan to load programs into Sorcerer or save programs stored in Sorcerer by recording them on a cassette tape.

There are two types of connectors at the back of Sorcerer for connection to cassette recorders (refer to Figure 1). Most of the time you will be using only one cassette recorder in manual operation, either playing a pre-recorded program into Sorcerer, or recording a customized program out. For those simple operations you will find two identical cassette recorder cables included with Sorcerer. Use them to connect the cassette recorder (MIC and EAR connectors) to the appropriately labeled MIC and EAR connectors at the rear of the Sorcerer.

If you want to conect two recorders, or have Sorcerer control your recorder's on-off switch, you can order a special data cable, Part No. DP4005. This cable attaches to the 25-pin connector labeled Serial Interface (refer to Figure 1).

It is important to adjust the cassette recorder for proper tone and volume setting. A general rule of thumb is a setting of half-on for both tone and volume. (Example: If your control knob is labeled 1 to 10, a setting of 5 would be appropriate.) Every recorder has different characteristics. The type of cassette tape used also effects the performance. Exidy data cassettes are recommended (Part No. DP3001) and are available as accessory items.

You can test the cassette tone and volume adjustments by writing and reading some test data. The following procedure uses the Sorcerer's operating system as a data source.

Depress the SHIFT LOCK key

Type: `DU E000 E01F` RETURN

(Be careful to include all the spaces.) Sorcerer will print a table of numbers and letters on the screen. Copy the body of the table on a piece of paper (don't bother with the data at the top and left side).

Now check that the two connecting cables are installed in the proper connectors of the recorder and Sorcerer. Insert a blank cassette into the recorder and rewind to the beginning of the tape. Press the RECORD and PLAY buttons on the recorder and wait five seconds.

Now type: `SA TEST1 E000 E01F` **RETURN**

After a few seconds the prompt (>) will appear; this means your data has been recorded. Rewind the tape.

Type: `LO TEST1 1 0000` **RETURN**

Press the PLAY button on the recorder. After about half a minute, this message should appear on screen:

```
FOUND - TEST1     0020 E000 0000

LOADING
```

Shortly after, Sorcerer should print some additional information and another prompt, to tell you that the data has been loaded back into the Sorcerer.

If you cannot get the FOUND message, or if you get a tape ERROR message, rewind the tape, re-adjust the recorder's tone and volume, and try again. When you get the data loaded into the Sorcerer,

Type: `DU 0000 001F` **RETURN**

Sorcerer will print a table of numbers and letters; compare this with the data you copied on paper.

If you have a Standard BASIC ROM PAC, you can test the cassette tone and volume by writing and reading a test program. Insert the ROM PAC as described above, and follow this procedure:

Depress the **SHIFT LOCK** key

Type: `NEW` **RETURN**

Type: `10 PRINT "THIS IS A TEST"` **RETURN**

`20 END` **RETURN**

To make sure the Sorcerer accepted your test program into memory:

Type: `LIST` **RETURN**

The program should appear on the screen.

Execute the program.

Type: `RUN` **RETURN**

The message THIS IS A TEST should be displayed on the screen. To save your program on a cassette tape proceed as follows:

Verify that the two connecting cables are installed in the proper connectors at the cassette recorder and back of Sorcerer. Insert a blank cassette into the recorder and rewind to the beginning of tape.

NOTE: Most cassette recorders have automatic level control (ALC) for recording and no adjustment is necessary to record programs.

Press the RECORD and PLAY push buttons on the recorder. Wait 5 seconds.

Type: `CSAVE TEST` **RETURN**

After a few seconds the message READY will appear on the screen. This means your program has been recorded.

Rewind the cassette tape.

Type: `NEW` **RETURN**

Type: `LIST` **RETURN**

Your test program should not print out on the screen. It has been erased by the command NEW.

Type: `CLOAD TEST` **RETURN**

Press the PLAY button on the recorder and after a few seconds the message READY will appear on the screen. Stop the recorder.

Type: `LIST` **RETURN**

Your test program should have loaded into memory and should now be displayed on the screen.

TRAVEL AIDS

(System Components and Introductory Programming)

By now you have discovered all the components that are required to make up a computer system. The Sorcerer is the main unit with the computational power of the system, but the system isn't complete without display and reasonable program storage capability. That is the purpose of a video monitor and cassette recorder.

With the video monitor attached to Sorcerer you have a functional computer. Even without the ROM PAC™ cartridge you can perform simple operations through keyboard commands, and communicate with Sorcerer through our ***Power-On Monitor.*** This operating system is very limited, however, and you will want use the ROM PAC cartridges to expand the power and usefulness of Sorcerer.

SOFTWARE — Computer programs or lists of instructions that tell the computer what to do. Usually on cassette tape, permanent memory or printed in the form of a listing.

A detailed description of the Power-On Monitor program is given in the section titled SCHEDULED PROGRAM.

The ROM PAC™ cartridges are many and varied. The software or programs provided in the cartridge are what we in the computer world refer to as ***systems software.*** Systems software is a measure of the power flexibility and usefulness of the computer. It is the systems software that communicates with the various input and output attachments of the Sorcerer (which could number as many as 256). The systems software also allows the user — you — to program in higher level languages.

> ***HIGHER LEVEL LANGUAGE — A means of communication with computers at a level of understanding much like conversation with another person.***

By programming in higher level language, you can instruct the machine to perform operations by use of word statements and commands, rather than cumbersome machine instructions.

Higher Level Language Example in BASIC

Line Number	Statement
1	PRINT "Do someting for me"
2	GO TO 1

Computers perform instructions in sequence; thus, we type a line number for each operation and then the instruction we want executed. In the example, PRINT and GO TO are BASIC language statements. If you have correctly inserted a Standard BASIC ROM PAC into the Sorcerer, these commands will cause the Sorcerer to print the message within the quote marks on the video monitor and then execute the next line instruction. In this case the GO TO 1 instruction sends the Sorcerer back to printing the message again.

If you were to type the example program into Sorcerer following each line with a RETURN , you would find the computer in an endless loop.

If you want the program to pause, or the screen display to stop scrolling, press the RUN/STOP key. Release the RUN/STOP key, and the program or display will continue. (The ESC key will do the same thing.) To stop a program or listing completely, press the CTRL and C keys simultaneously. When you stop a program this way, Sorcerer will print a message telling what line of the program it stopped at.

Before you begin writing programs, it is helpful to have some knowledge about programming in general. We all use manual programs to make everyday decisions. You might say programming is as common as getting out of bed in the morning, driving to work or planning your retirement.

Let's explore programming for a few minutes with a look at ***your*** internal program. You have a lot in common with the fundamentals of computing. Your microcomputer is your brain with memory and muscular/nerve system for input/output control.

Let's Compare Your Computer with Sorcerer . . .

FUNCTIONS	EXIDY SORCERER	YOU
Central Processing Unit (CPU)	Z80	Brain
Program Storage	Memory (ROM, RAM, Cassette)	Mind
Interface	I/O Circuitry	Nerve System
Input	Typewriter Keyboard	Eyes, Ears, etc.
Output	Video Display	Mouth, Hands, etc.

As you can see, computers have been around for a long time, in one form or another; so has computer programming. A program is just a sequence of operations or instructions designed to solve a problem or perform a task.

Let's Examine a Typical Program Sequence

Problem: It is getting late, you are tired, you want to go to sleep.

Solution: Go to bed.

STEP	OPERATION
1	Secure Household
2	Go To Bedroom
3	Prepare for Bed
4	Set Alarm Clock
5	Get Into Bed
6	Close Eyes
7	Count Sheep
8	Sleep
9	End

Of course you don't think of each program step as you perform it, because it's something you do every day and by now is second nature. However, when you plan to do something new or different, you give it a little thought or preparation (programming) before you do it (execution). Sometimes it's easier to determine the best direction or sequence by looking at an overall view of what you want to accomplish. In computer talk this procedure is called writing a ***flow chart*** of your program. Flow charts use simple geometric outlines for each step in a program as shown in Figure 2.

Flow charts help you visualize a problem and organize the best program solution. They also let you explain your program to someone without going into great detail.

The next step is to break down the operations to specific instructions which your computer understands. Your Sorcerer can communicate in many different languages, and you can change languages just by changing program cartridges. This provision is important because languages are more efficient or convenient in some applications than in others.

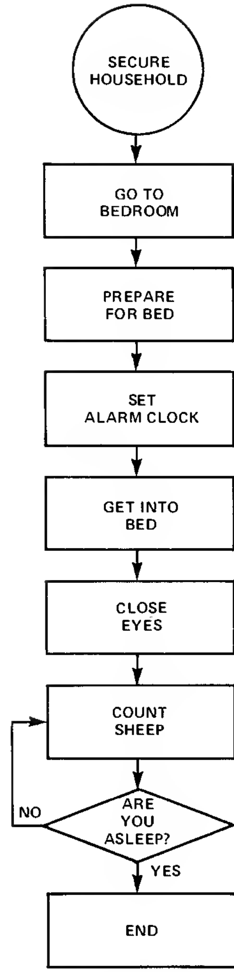


Figure 2. Program Flow Chart

You don't have to know how to program to use your Sorcerer because many programs are readily available on inexpensive tape cassettes.

BASIC is considered to be the best language for educational applications; FORTRAN (derived from the words FORmula TRANslation) is, as you might suspect, a good scientific language; COBOL is a business oriented language, and the list goes on.

Application programs are the functional programs of computing. They are, as the name implies, programs that are applied to perform tasks. Many application programs have been written for game playing, bookkeeping, education, etc. and Exidy provides these programs on cassette tape to keep the expense of your program library in line with your budget for personal computing. Cassettes are much less expensive than program cartridges and many different programs may be recorded on one cassette.

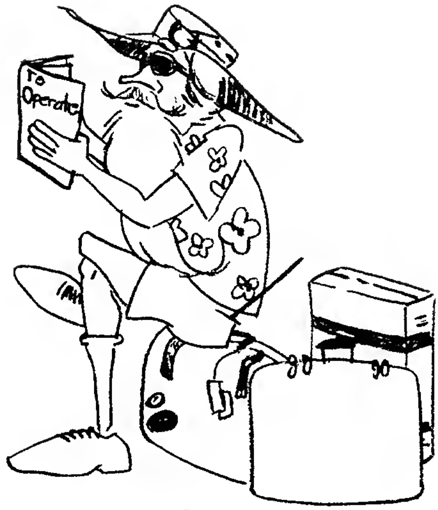
You will even find many application programs published in personal computing magazines. These can be typed directly into Sorcerer for recording on cassette, if written in Standard BASIC or the program language that corresponds with the ROM PAC cartridge inserted in the Sorcerer.

While you are learning how to program, it is always helpful to look at a program written by someone else, for examples in technique. You do this in Standard BASIC by typing the LIST command after putting an application program in memory. This is discussed in detail in the BASIC Tour manual which accompanies the ROM PAC cartridge.

There isn't enough space in this book to teach you to program in all the languages; there are too many of them, and new ones are being developed all the time. However, a complete instruction manual is included with each ROM PAC.

BON VOYAGE

(Operation and Keyboard Description)



Before you embark on your tour, the Sorcerer will provide escort service so that you become familiar and comfortable with your new surroundings.

Test the keyboard by typing in your name. If you type a wrong character, you may erase it by depressing SHIFT RUB. Then proceed as you were. Don't worry about making mistakes — you already know how to erase them. Your Sorcerer will also be looking for mistakes. Since your name is not a legal command, Sorcerer would display an error message if you were to press RETURN following your name.

> ***Remember, you can't damage the Sorcerer with keyboard operations . . . you may confuse Sorcerer, but you can always re-establish communications by simultaneously pressing two RESET keys.***

You can order Sorcerer to automatically solve a problem hundreds or even thousands of times, with new information each time. Since the computer can run through hundreds of program steps in a second, it's ideal for long and repetitious problem solving.

You must understand the keyboard if you are to have any meaningful conversations with Sorcerer.

The keyboard is similar to a standard typewriter keyboard, with a few additions. The letters and control keys are virtually in the same positions, but for your convenience and speed in numerical computations, a numeric pad is provided (see Figure 3).

Figure 3. The Keyboard (Characters without Shifting)

Since your Sorcerer has both upper and lower case characters, you will find its operation identical to a typewriter. And, since many programs are written only in upper case characters, there is a shift lock key for your convenience.

If you press a key without shifting, you will get a lower case character.

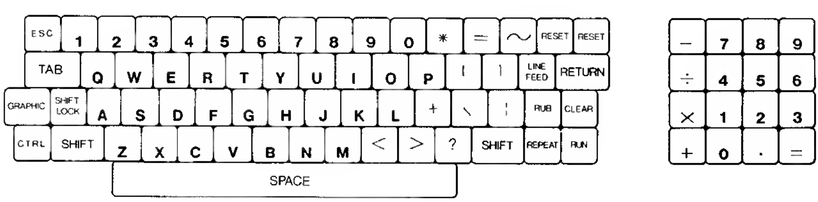


Figure 4. Characters with Shift Lock Key Depressed

When you simultaneously press the SHIFT LOCK key and a letter, you get the upper case representation or notation as shown in Figure 4.

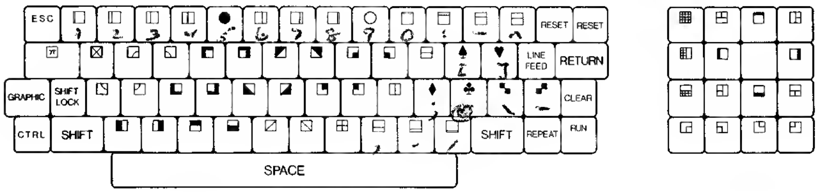


Figure 5. Characters with Graphic Key Depressed

When you simultaneously press the GRAPHIC key and a letter you get the graphic symbol shown in Figure 5.

The graphic symbols shown on the key tops are a special defined set for Sorcerer. They are used to draw pictures, lines and bar charts on the screen. They can be used just like any other letter or digit on the keyboard.

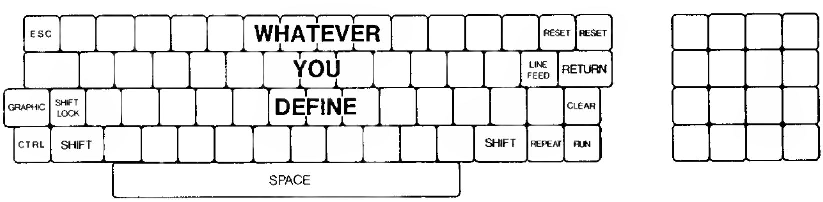


Figure 6. Characters with Shift and Graphic Keys Depressed

Pressing SHIFT and the GRAPHIC key simultaneously with any letter will give you one of your own characters. (See the SPECIAL OFFERING chapter for detailed instructions on how to program your own character set.)

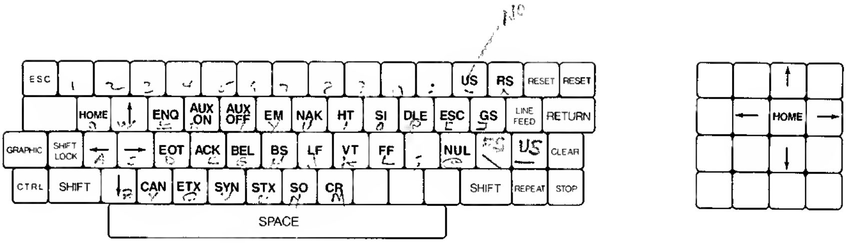


Figure 7. Characters with Control Key Depressed

Figure 7 shows how you can perform the standard TTY and computer terminal functions, by pressing the CTRL key simultaneously with the designated keys. These functions correspond to ASCII characters 0 through 31. If you are new to personal computing, you can ignore these completely.

Figure 7 also shows which keys control the video monitor cursor. With the CTRL key depressed, either of the HOME keys will put the cursor in home position (upper left corner of the screen), while an arrow key will move the cursor 1 space in the direction of the arrow.

EXPERIENCED TRAVELER

(Scheduled Program)

ALL SORCERERS ARE CREATED EQUAL.

The basic (meaning fundamental) intelligence of your Sorcerer is a program called the **Power-On Monitor**. Should you ever get lost, confused, even bewildered, an understanding of the Power-On Monitor will prove to be a useful tool in regaining your composure and discovering where your problem lies.

The Monitor program is buried inside the Sorcerer and has nothing to do with program cartridges. It does, however, look to see if a program cartridge is installed in the unit and if so, it will pass control to the cartridge. When a program cartridge isn't installed, the Power-On Monitor program will take control and print >.

Hint . . . if you have a program cartridge plugged into Sorcerer and a > is displayed on the screen, you don't have the cartridge inserted properly.

The greater than [>] character is referred to as a ***prompt;*** it lets you know that Sorcerer is awaiting a command from you. The underscore character [_] is called a ***cursor*** and tells you where the next character will appear on the screen.

The Monitor program can be called upon at any time. Not everyone will want to escape into the Monitor because it isn't nearly as fun as BASIC or other higher-level languages. Its main purpose in life is to diagnose hardware and software programs. For those of you who want that challenge, here is a list of Monitor commands.

MONITOR COMMANDS

The commands are listed in the left column; only the first two letters are necessary. The ***parameters*** of each command are listed to the right; parameters enclosed in parentheses are optional. You must use spaces or other delimiters between the command and its parameters, and between parameters. Follow the last parameter with a RETURN.

DUMP XXXX (XXXX)
addr addr

Displays memory on the screen. If you give one address, you get the contents of that address; if you give two addresses, you get the contents of all addresses from the first to the second.

Examples:

DU 0100	Displays the contents of address 0100.
DU FE00 FE07	Displays the contents of all address from FE00 to FE07.

ENTER XXXX
addr

Lets you enter hexadecimal numbers into the memory address. When you hit the carriage return, Sorcerer advances to the next address; when you hit carriage return alone, you just get the current address. A / (slash), followed by the carriage return, gets you out of ENTER.

Examples:

Entering data one address at a time:

EN FE00	The command to enter data, starting at FE00.
FE00: FF	Sorcerer prints the addresses and colons;
FE01: 00	You must enter the numbers (following each with a RETURN).
FE02: FF	
FE03: 00	
FE04: /	The slash terminates the EN command.

Entering a block of data:

EN FE04	The command to enter data, starting at FE04.
FE04: FF 00 FF 00	Sorcerer prints the address and colon; you must enter the numbers, separated by delimiters (in this case, spaces).
FE08: /	Sorcerer reports the next available address; the slash terminates the EN command.

SAVE NAMEX XXXX XXXX (X)

NAMEX	XXXX	XXXX	(X)
name	addr	addr	unit

Saves memory on tape from the first address to the second address, on the tape unit specified (or on unit #1, if you don't specify). The header on the tape will contain the name (which must start with a letter), the beginning address, block size, file type, and GO address.

Example:

SA TEST1 0100 0240 2	Saves the program in addreses 0100 to 0240 on tape unit #2, under the name TEST1.

LOAD(G) (NAMEX) (X) (XXXX)
name unit addr

Loads the file named NAMEX into memory, from the tape unit specified (or unit #1, if you don't specify). Starts loading at the given address if specified; otherwise at the address in the file header on the tape. If you specify an address you must also specify the unit. If you don't give a name, then the first file will be loaded. If you include the G after LO, Sorcerer will run the program starting at the GO address in the file header, immediately after loading. This command will only load files saved in the Monitor. A file saved in Standard BASIC must be loaded with the command CLOAD; a file saved in the Monitor can be loaded with CLOAD.

Examples:

LO TEST1	Loads the program TEST1 from tape unit #1, starting at the address in the file header.
LO TEST1 2	Loads the program TEST1 from tape unit #2, starting at the address in the file header.
LO TEST1 1 0100	Loads the program TEST1 from tape unit #1 starting at address 0100.
LOG	Loads the first program on tape unit #1 starting at the address in the file header, and runs the program.

	LOG 2 0100	Loads and runs the first program on tape unit #2, starting at address 0100; then jumps to the GO address in the file header.
	LOG TEST1	Loads and runs the program TEST1 from tape unit #1, starting at the address on the file header.

FILES (X)
unit

Lists all cassette files and header information from the specified unit. CTRL C stops the listing and returns you to the Monitor. This command will list BASIC programs stored with CSAVE, but will ***not*** list arrays saved with CSAVE * .
Verifies program on tape for CRC errors. Can be used to check tape without altering memory.
Terminate = CTRL C

GO XXXX
addr

Calls a program at the given address.

MOVE XXXX XXXX (S)XXXX
addr1 addr2 block/addr3

Copies the contents of a block of successive memory addresses into another block of addresses. Use this command two ways:

1. If you include the S, then the following number tells Sorcerer how many addresses to move. The contents of address 1 go into address 2, and the addresses following address 1 are copied after address 2.

2. If you omit the S, then Sorcerer moves the block of memory starting at address 1 and ending at address 2. The contents of address 1 go into address 3, and the addresses following address 1 are copied after address 3.

Examples:

MO FC00 FE00 S0010	This copies the 16 addresses (10 Hex) starting at FC00, into the 16 addresses starting at FE00.
MO FC00 FC0F FE00	This copies the 16 addresses from FC00 to FC0F into the 16 addresses starting at FE00.

TEST

XXXX XXXX (C)
addr1 addr2

Tests each bit of RAM from address 1 to address 2 (8 bits per word). A blinking * appears while the test is running. No message if memory is good; otherwise prints BAD or OK for each bit. With C, tests continuously, giving a PASS COMPLETED message at each pass. Use this command only for testing RAM; all ROM addresses will be reported BAD, no matter what their condition.

NOTE

The bit test (command TE) will not give reliable results on the area of RAM occupied by the Monitor stack. If you wish to test this portion of RAM, first relocate the Monitor stack, as described under that heading. For the addresses of the stack, see Figure 9, Appendix A.

You must also take special precautions when testing screen RAM and video scratch RAM (addresses F000 to F7FF. For details, refer to the Sorcerer Technical Manual.

PROMPT

=X

Changes the Monitor prompt from > to whatever you specify. For example, PR = # changes the prompt to #.

CREAT

Creates a batch tape on tape unit 1. Enter your batch commands a line at a time after the * prompts, and end each line with a carriage return. The tape will start, record your line, and stop. To reenter the Monitor, hit the carriage return alone.

LIST

Lists all the commands on your batch tape.

BATCH

Tells Sorcerer to read the batch tape on tape unit 1 and execute all the commands on that tape. You can stop the routine by typing CTRL C ; Sorcerer will stop automatically if it reads an invalid command, or if any error occurs.

OVER

This is the batch command which terminates the batch mode and returns you to Monitor. This is similar to the END statement in Standard BASIC.

SET

41 - EXEC (1200)
4D - MASTER (300)

S = XX	Changes display delay to XX.	
T = X	Sets tape rate to 0 = 1200 baud (default); 1 = 300 baud.	
F = XX	Sets file type in tape header to ASCII value of XX. If D8 is set, then file is non-auto-execute.	
X = XXXX	Sets auto execution address in tape header.	
O = V	Sets current output port to	VIDEO
= P		PARLOT (parallel output)
= L		CENTRONICS PRINTER DRIVER
= S		OUTAPE (tape output)
= XXXX		ADDRESS
I = K	Sets current input port to	KEYBRD
= P		PARLIN (parallel input)
= S		INTAPE (tape input)
= XXXX		ADDRESS

NOTE

There are 256 file types (00 through FF). D8 is non-auto-execute. All others are arbitarily user assignable. However, BASIC will only load a file of type BX (i.e., most significant digit is B).

Examples:

SE S = FF	Sets the display delay to FF.
SE F = D8	Sets file type to non-auto-execute.
SE O = L	Sets output to Centronics Printer.
SE O = 1000	If you have loaded an output driver routine at address 1000, this command sets the Sorcerer's output to that routine.

PP (X)

Jumps to ROM PAC. If no parameter, then warm start, otherwise cold.

NOTE: Each address is a four digit hexadecimal (base 16) number, from 0000 to FFFF (0 to 65535 in decimal notation). The content of each address is a two digit hexadecimal number, from 00 to FF (0 to 255 decimal).

Until the list of Monitor commands came along your tour of personal computing was pretty civilized. In this chapter you're about to have a close encounter with the stuff the computer world is famous for . . . abbreviations, mneumonics, hexadecimal notation and never enough explanation in the instructions. Never fear . . . Your Sorcerer will help you through all this because if you don't understand an instruction ***experiment*** until you get a resonable response.

Let's go back and look at that list again. DU is an abbreviation for "dump memory," and in fact that is exactly what happens if you type DU followed by an address that you want to display, followed by a carriage return. The problem now is to understand this memory address

notation. Each digit of the four shown is a hexadecimal (base 16) digit. Hexadecimal is a number system often used by computers. It uses digits 0 through 9 and the letters A through F. The letters are used because it is necessary to represent 16 different values with a single digit for each value; the letters A through F represent the number values 10 through 15.

Using letters in counting may appear awkward until you become familiar with the system. Hexadecimal address 1E equals the decimal address 30.

DECIMAL		HEXADECIMAL	DECIMAL		HEXADECIMAL
0	=	0	9	=	9
1	=	1	10	=	A
2	=	2	11	=	B
3	=	3	12	=	C
4	=	4	13	=	D
5	=	5	14	=	E
6	=	6	15	=	F
7	=	7	16	=	10
8	=	8	17	=	11

When conversation turns to talk of computer address and instructions you will always find yourself with foreign number systems. Most likely binary (base 2), octal (base 8), hexadecimal (base 16) and not often enough, decimal. For this reason you will find conversion tables in many computer publications. Since IBM is partial to hexadecimal you will find our reference to address and instructions in hexadecimal also. Computer instructions in hexadecimal are also referred to as ***machine language*** code which is described in great detail in the manual that accompanies the development program cartridge.

MONITOR NOTES

1. Monitor program resides from E000 HEX to EFFF HEX.

2. Video resides from F000 HEX to FFFF HEX.

3. Power Up —

The Monitor searches and tests memory from 0000 HEX to the end of memory. It then places its stack approximately 128 bytes below the top of RAM. If a ROM PAC is inserted, the monitor will then transfer to it. Otherwise, it will then print:

```
EXIDY STANDARD MONITOR

VERSION 1.0
COPYRIGHT (C) 1978 BY EXIDY INC.

THE TOP OF RAM IS XXXX HEX.
STACK BEGINS FROM XXXX HEX.
```

4. Entering the Monitor —

The Monitor has three (3) entry points:

COLD
WARM
USER

E000 — COLD —	is always entered from power on. The RAM is searched and the top is found. The video is initialized and default parameters are set up.
E003 — WARM —	establishes a stack from information from COLD or USER and re-enters the Monitor.
E006 — USER —	uses the HL register pair as the top of RAM and then jumps to WARM.

5. I/O Entry Points —

The Monitor has nine (9) I/O entry points:

E009 — RECEVE —	gets character from current input device to the A register; if no character, then sets the A register to zero. If character is present, Z flag is reset; otherwise, Z flag is set.
E00C — SEND —	outputs the A register to current output device.
E00F — INTAPE —	inputs to the A register from tape.
E012 — OUTAPE —	outputs the A register to tape.

E015 — QUICKCK — does a quick check for CTRL C , ESCAPE or RUN/STOP . Returns with Z set if none.

E018 — KEYBRD — inputs to the A register from the keyboard, if a key was pressed. Z flag is reset or set, depending on whether a key was pressed.

E01B — VIDEO — Outputs the A register to screen

CTRL W = cursor up
CTRL A = cursor left
CTRL S = cursor right
CTRL Z = cursor down
CTRL H = character delete
CLEAR or form feed = clear screen
CTRL Q = cursor home

> **NOTE**
>
> The form feed also resets the standard graphics to their normal values (the symbols shown on the key caps).

E01E — PARLIN — inputs to the A register from parallel input port.

E021 — PARLOT — outputs the A register to a parallel output port.

6. Delimiters —

The Monitor will use any of the following to delimit or to space between Monitor parameters on a line:

SPACE ! " # $ % & ' ( ) * + , – .

7. Error Messages —

The Monitor has three error messages:

```
INVALID COMMAND
INVALID PARAMETER
TAPE CRC ERROR
```

The first two are self explanatory; these are syntax errors and often result from incorrect use of delimiters. A tape error message usually indicates incorrect tone or volume settings of the tape recorder, but may also indicate a malfunction in the Sorcerer's tape interface or in the tape cartridge itself.

THE MONITOR BATCH SYSTEM

The Monitor batch system lets the Sorcerer automatically run a sequence of files. The four batch commands are explained above (see Power-On Monitor Command Set).

A ***batch tape*** is a sequence of user created Monitor commands, recorded on a tape cassette. Use the Monitor command CR to create this tape, and the command LI to verify its contents. Use the command BA to automatically execute the commands on the batch tape, and use the command OV as the last command on the tape, to return control to the Monitor. Note that the monitor does not have any commands which would allow you to edit a batch tape. If you make a mistake while creating your tape, you will have to go back and start over.

Example:

Here is a sample batch routine. The following commands are on a batch tape ready to be played on tape unit #1. Tape unit #1 also contains the two programs PROG1 and PROG3. Tape unit #2 contains a blank tape, for recording the file FILE1.

LOG PROG1 1
PROG1 is here

LO PROG3 1 2000
PROG3 is here

SE O = L

GO 2000

SE F = AA

SA FILE1 0000 1000 2

OV

This batch routine loads the program PROG1 from tape unit #1 and executes it. Next, it loads the program PROG3 from tape unit #1 starting at address 2000. It then routes the output to a Centronics printer and executes PROG3. Then it sets the tape file type to AA and saves the memory from addresses 0000 to 1000 on tape unit #2, under the name FILE1. Finally, it returns control to the Monitor.

Any error will end the batch mode and return you to the Monitor. You can also return to the Monitor by hitting CTRL C, ESC, or RUN/STOP.

Only tape unit #1 can be used for creating or executing a batch tape; however, you can put a sequence of programs on tape unit #2 and run them with batch tape LOG commands. To use both tape units, you need a Serial Data Cable (Exidy Part No. DP4005); this cable also provides automatic control of the tape unit motors.

We strongly recommend you use the Serial Data Cable — without automatic motor control, your batch tape may run past a command before the preceding command has been executed. If you do not use the data cable, you can avoid this problem in two ways:

First, you can put plenty of blank tape between commands on the batch tape. When you enter each command (under the CR command), let the tape run awhile before hitting RETURN.

Second, you can start and stop the tape unit manually (although this defeats the purpose of the batch system). The Sorcerer prints each command on the screen after reading it from the batch tape; this is the time to stop the recorder. After executing the command, Sorcerer prints a prompt; you can then turn on the recorder.

The Monitor puts your commands onto a batch tape in this format:

One hundred 0s and a 1	a command in ASCII	one byte CRC (Cyclic Redundancy Check)

As each character is written on (or read from) the batch tape, Sorcerer performs a CRC routine to update the CRC byte. The final value of this byte is then written on the tape (under the CR command) or compared with the value already on the tape (under the BA command).

RELOCATING THE MONITOR STACK

The stack is a term which loosely denotes three separate areas of memory in the Sorcerer:

- The Monitor RAM (temporary scratchpad storage for the Monitor)
- The Monitor stack proper (temporary storage for Z80 registers and return locations of CALLs)
- The stack pointer (a two byte register in the CPU)

To ***relocate the stack*** means to simultaneously move the Monitor RAM and stack proper, and reset the stack pointer.

At power-on, Sorcerer searches RAM for the top RAM address. The 112 addresses (70 hexadecimal) from the top downward are used as the Monitor RAM; the next 64 addresses approximately (40 hexadecimal) are the Monitor stack proper.

Example:

In the 8K Sorcerer at power-on, the top of RAM is 1FFF and the stack starts at 1F90. This means that addresses 1F91 to 1FFF are Monitor RAM storage, and 1F90 to approximately 1F50 are the Monitor stack proper.

If the stack is disturbed, the system may crash. This usually happens when a tape file or user program overwrites the stack. You can recover from the crash by hitting the RESET keys, but you will still lose the contents of all RAM. To prevent a crash, you must relocate the stack to an area of RAM which won't be used by your program or tape file. This area must contain at least 176 bytes (B0 hexadecimal) — 112 bytes for Monitor RAM storage, and 64 bytes for the Monitor stack proper.

You must also relocate the stack before using the Monitor RAM test (command TE) on the area of RAM occupied by the stack.

To relocate the stack, first, choose a suitable address XXYY for the top of the Monitor RAM; here, XX and YY are the high order and low order bytes of the address, respectively. Second, use the Monitor EN command to put these Z80 instructions into the addresses 0000 to 0005:

21 YY XX C3 06 E0

This Z80 program loads the address XXYY into the HL register pair and then jumps to the Monitor USER entry point. Finally, give the command GO 0000.

Example:

To move the stack so that the top of RAM is at 0750:

You type:	EN 0000
Sorcerer replies:	0000:_
You type:	21 50 07 C3 06 E0 /
Then type:	GO 0000

Sorcerer moves the stack and prints this message:

```
EXIDY STANDARD MONITOR

VERSION 1.0
COPYRIGHT (C) 1978 BY EXIDY INC.

THE TOP OF RAM IS   0750 HEX.
STACK BEGINS FROM   06E1 HEX.
```

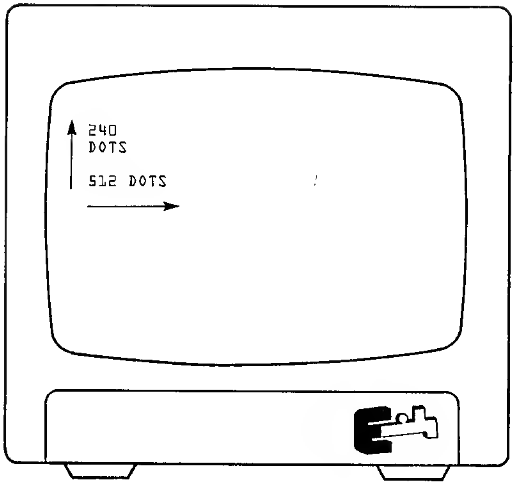
SPECIAL OFFERING

(User-Defined Graphics)

A picture is worth a thousand words and you're about to discover how to create your own pictures with the Sorcerer graphics. In the BON VOYAGE chapter you learned about the standard keyboard graphic characters, which you can type directly onto the screen.

Let's look at the graphics more closely. Your Sorcerer has a graphic resolution of 512 x 240; that is to say, there are 512 horizontal positions and 240 vertical positions for dots to appear on the screen. To print a character on the screen, Sorcerer selects an 8 x 8 square of these positions and puts dots in just the places needed to form the character.

Since each character occupies an 8 x 8 space, and the screen is 512 x 240, it follows that you can get 64 characters on a line, and 30 lines on the screen. The total number of characters on the screen at any time (including blank spaces) is 1,920.



Now the question is who gets to say what's on the screen? Using the SHIFT and GRAPHIC keys in combination with the keyboard you have access to 256 different characters. It's only fair that you be able to define as many characters as the Sorcerer so we have a compromise. 128 fixed characters (Standard ASCII) and 128 programmable.

THE COMPROMISE IS THIS:

SORCERER VS.	YOU
Since the keyboard needed to be labeled and most of the people in the world relate to typewriters, Sorcerer selected the upper and lower case alphanumeric character set and you can't change those, even if you don't speak Sorcerer's language. One-hundred twenty-eight characters you can't change.	Since there was room on the keytops, Sorcerer has assigned 64 of the 128 characters reserved for graphics. But, in this case you can change those if you want. You, too, have control of 128 characters and if you don't like Sorcerer's selection feel free to change them at will, even into your own language if you don't like Sorcerer's; after all, they don't have to be graphic symbols.

To Create Graphic Characters

Now that you know what you can do, it's only fair that you know how. Each character is stored in eight successive memory addresses in the memory block FE00 to FEFF (hexadecimal). For example, the first key (the 1 key on the main keyboard) uses FE00 to FE07. So first you have to figure out which eight addresses go with the key in which you want to store your new character.

Your character will be represented as an 8 x 8 array of dots and blanks, so get out a pencil and paper, and work out the representation. Quarter-inch grid graph paper is useful here. Each eight-place row of this array will fit in one of the eight addresses of your character's key; the top line goes into the first address, the next line goes into the second, and so on.

Example:

One way to represent the Greek letter omega (Ω) is:

```
■■■■■■■■
■■□□□■■■
■□■■■□■■
□■■■■■□■
□■■■■■□■
■□■■■□■■
■■□■□■■■
□□□■□□□■
```

Now take each line of your array, and change it into a two-character code in this way: First split the eight place line in two. Then convert each of these two four-place lines into code with the following table:

Four-Place Line Code

■■■■	=	0	□■■■	=	8
■■■□	=	1	□■■□	=	9
■■□■	=	2	□■□■	=	A
■■□□	=	3	□■□□	=	B
■□■■	=	4	□□■■	=	C
■□■□	=	5	□□■□	=	D
■□□■	=	6	□□□■	=	E
■□□□	=	7	□□□□	=	F

Examples: The eight-place line:

□□■■□■□■ is coded as CA

The eight lines of our letter omega are:

■■■■■■■■ 00
■■□□□■■■ 38
■□■■■□■■ 44
□■■■■■□■ 82
□■■■■■□■ 82
■□■■■□■■ 44
■■□■□■■■ 28
□□□■□□□■ EE

What you have just done is to consider each eight-place line of your character as an eight-digit binary number (white dots are 1s and blanks are 0s), and to re-write that number in hexadecimal notation. So you now have a two digit hexadecimal number for each of the eight lines of your new character; you must now get these numbers into the eight memory locations that belong to your character's key.

Get into the Power-On Monitor. Then give the Monitor command EN XXXX where XXXX is the first memory address of the desired key. For example, the first address of the **2** key on the main keyboard is FE08, so the proper command for that key is EN FE08. Sorcerer will respond with the address you gave it, followed by a colon.

Type in the two-digit hexadecimal code of your character's first line, and hit **RETURN** . Sorcerer will reply with the next address in memory, and wait for more data. Type in the hex-code for your character's second line, and repeat this process until you have entered all eight lines of your character into memory. Sorcerer will then give you the address of the first memory location for the next key on the keyboard. Type in a slash (/) and hit **RETURN** .

Your new character is now stored in the Sorcerer. Just press **GRAPHIC** and **SHIFT** and your character's key, and your new graphic will appear on the screen.

Note: Your special graphic characters normally live in memory addresses FE00 to FFFF. Addresses FC00 to FDFF contain the standard

graphic characters shown in Figure 5 on page 21; these are the characters you get using the GRAPHIC key, but not the SHIFT. You can put your own characters into these addresses if you wish; the procedure is the same as above.

ALSO <CLEAR>

When you turn off Sorcerer's power, or hit RESET all your user-defined characters will be lost. To save them for future use, pass to the Power-On Monitor and save the contents of the character's memory addresses on cassette, using the Monitor command SA. You can load the characters back into memory from the tape, using the Monitor command LO.

Figure 8 shows the memory addresses and ASCII numbers of user-defined characters. The four digit hexadecimal number at the top of each key is the first of the eight memory addresses which store the character. The two hexadecimal numbers at the bottom of the key are the ASCII numbers of the standard graphic and user-defined graphics of that key.

Example:

The S key on the main keyboard controls the user-defined graphic stored in addresses FED8 to FEDF. The ASCII number of that user defined graphic is DB. The ASCII number of the S key's standard graphic is 9B.

FECC
80/C0
address (hex)
ASCII #
c
GRAPHIC
ASCII #
c
SHIFT/GRAPHIC

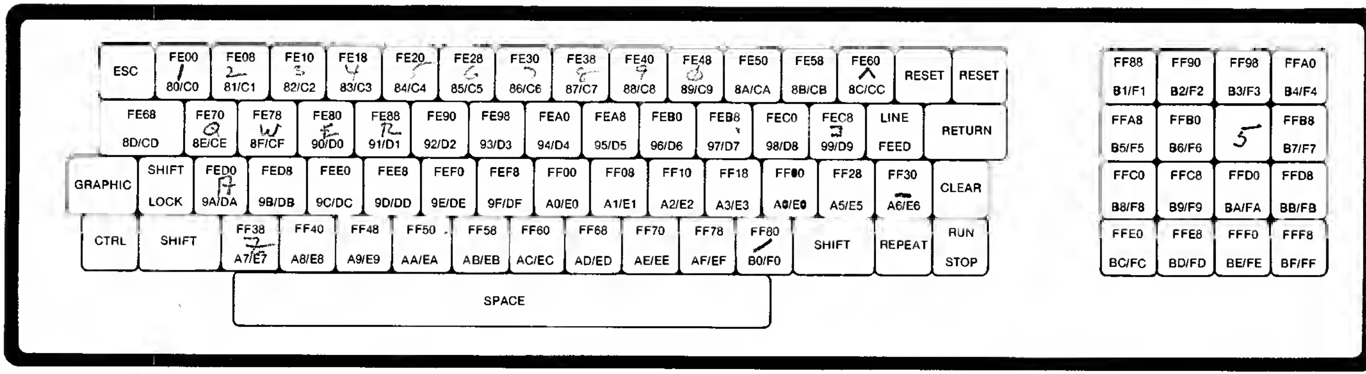


Figure 8. User Graphic Storage Addresses and ASCII numbers

FURTHER INFORMATION

(Serial/Parallel Ports, Cassette I/O, Video, S-100 Bus)

Until now we have described the Sorcerer as a complete computer system having the fundamental ingredients of central processor, memory, keyboard input, video display output, and cassette tape for program storage and playback. The actual hook-up or installation of the computer system was relatively easy and with A.C. power applied to all the units, the operation was automatic with a small degree of prompting.

As you have seen, life with the Sorcerer is more interesting when you write your own programs (or customize someone else's). You will find that customizing your system's hardware can be just as much a challenge. Therefore, we should prepare you for further exploration in the hardware domain with an explanation of the built-in electronic interfaces.

PLUG-IN CARTRIDGE

The ROM PAC Cartridge, even though it can be plugged in and out, is just an extension of the computer's main memory; therefore, it does not qualify as an I/O interface and we shall say no more about it.

KEYBOARD

Your keyboard is definitely an I/O device. Specifically, it is an input device. However, it is totally dedicated to your communication with the Sorcerer and you cannot customize it, so enough said.

VIDEO DISPLAY

The video is also dedicated to your communication with Sorcerer but a certain degree of flexibility is desirable. A standard composite video signal is provided in the video connector at the back of Sorcerer for attachment to the display unit of your choice. The range of video display units varies from 5" to 23" with many popular sizes in between. Your choice is a toss-up between character size displayed and your eyesight.

CASSETTE TAPE

The cassette interface is really your first encounter with programmable I/O. You can connect a second tape drive and implement the computer-controlled motor on/off. As you recall, the simple cassette operation just entailed plugging the two cables provided to the MIC and EAR phono jacks for cassette recorder #1 and manually pushing the play and record buttons. If automation is desired, Table 3 provides the programming data and command instructions and Table 4 provides the electrical connection information.

LINE PRINTER

Sorcerer has a Parallel Data Port, which we have not yet investigated. This data port is not essential to the normal operation of your Sorcerer, but you will need to know about it if you want to print your output on paper. When you add a hardcopy printer to your system, you will attach it to the left hand 25-pin connector at the rear of the Sorcerer; Table 2 provides the programming data and command instructions and Table 5 provides the electrical signal information.

REMOTE TERMINALS

The Sorcerer can be used as a smart terminal in applications where communication to another computer large or small is desired. This feat is accomplished with the serial I/O interface provided by the right hand 25-pin connector at the rear of Sorcerer. Attaching a data cable to this connector and either a modem or acoustic coupler enables you to transmit and receive information via telephone lines. The serial I/O is RS232 level compatible with data transfer rates of either 300 or 1200

baud. The baud rate is software selectable and the programming data and command instructions are provided in Table 1. The electrical signal information is described in Table 4.

EXPANSION

The slot at the rear of Sorcerer allows connection of the Exidy Expansion Unit. You will notice a 50 pin printed circuit edge connection on the main P.C. board. This is mated to a 50 pin edge connector and flat cable for connection to the S-100 bus translation card in the Exidy Expansion Unit.

Table 1. Serial Data Port Programming Assignment

I/O Port #	Bit #	Input	Output
FC	0 1 2 3 4 5 6 7	input data bits	output data bits
FD	0	transmit buffer empty	bits per character (NB1)
	1	data available	bits per character (NB2)
	2	over-run	number of stop bits
	3	framing error	parity select
	4	parity error	no parity

Table 2. Parallel Data Port Programming Assignment

I/O Port #	Bit #	Input	Output
FF	0 1 2 3 4 5 6 7	input data bits	output data bits
FE	6		data acknowledged
	7	data available	

Table 3. Cassette Recorder Programming Format

I/O Port #	Bit #	Input	Output
FC	0 to 7	input data bits	output data bits
FD	0	transmit buffer empty	
	1	data available	
FE	4		motor control #1
	5		motor control #2
	6		baud rate (0 = 300 baud, 1 = 1200 baud)
	7		interface (0 = cassette, 1 = RS232)

Table 4. Serial Interface (Right Hand 25 Pin Connector)

Pin #	Signal	Pin #	Signal
1	Shield 1	13	Motor #2+
2	RS232 OUT	14	Shield 2
3	RS232 IN	15	Mike 1
4	Ground	16	Mike 2
5	Aux 1	17	Ground
6, 7, 8	Ground	18	Aux 2
		19	Ground
		20	Ear 1
9	+12 volts	21	Ear 2
10	Unused	22	Unused
11	RS232 IN	23	RS232 OUT
12	Motor #1+	24	Motor #1—
		25	Motor #2—

Table 5. Parallel Interface (Left Hand 25 Pin Connector)

Pin #	Signal	Pin #	Signal
1	Ground	13	Input Bit 6
2	Output data accepted	14	Unused
3	Output data available	15	+5 volts
4	Output bit 7	16	Output bit 0
5	Output bit 6	17	Output bit 1
6	Output bit 5	18	Output bit 2
7	Output bit 4	19	Output bit 3
8	Ground	20	+5 volts
9	Input data available	21	Input data accepted
10	Input bit 0	22	Input bit 1
11	Input bit 2	23	Input bit 3
12	Input bit 4	24	Input bit 5
		25	Input bit 7

APPENDIX A

SORCERER'S MEMORY MAP

The Sorcerer will recognize up to 65536 different memory locations. These locations are given numbers from 0000 to FFFF hexadecimal (0 to 65535 decimal). Each memory location will contain one of the 256 hexadecimal numbers 00 to FF (0 to 255 decimal). If the Z80 CPU is instructed to read an address that is ***not*** connected to any RAM or ROM, it will usually assume the data is FF.

Figure 9 shows the functions served by each block of addresses. Column A shows the 8K Sorcerer with an 8K ROM PAC (such as Standard BASIC) inserted; the diagram is not to scale. Column B is the same as column A, but redrawn to approximate scale.

Columns C and D show alternative configurations. The lower portions show the addresses of internal RAM in the 16K and 32K Sorcerers, while the upper portions show the addresses of the 4K and 16K ROM PACs.

The unassigned addresses between the top of internal RAM and the bottom of the ROM PAC can be given to the S-100 Expansion Unit. If the ROM PAC is unplugged, its address space is also available to the S-100 unit.

The Monitor stack, consisting of the Monitor RAM and the stack proper, is shown at the top of the internal RAM, the position it usually takes at power-on or reset (see Relocating the Monitor Stack, for details). But, note that if a block of expansion memory is assigned immediately above internal RAM, the stack will go to the top of that block.

Example:

In the 16K Sorcerer at power-on or reset, the Monitor stack occupies addresses 3F50 to 3FFF. If 4K of memory (1000 Hex. addresses) is added to the Expansion Unit and assigned addresses 4000 through 4FFF, the Monitor stack will occupy 4F50 through 4FFF at power on. However, if the new memory is assigned to 5000 through 5FFF, the stack will remain 3F50 to 3FFF.

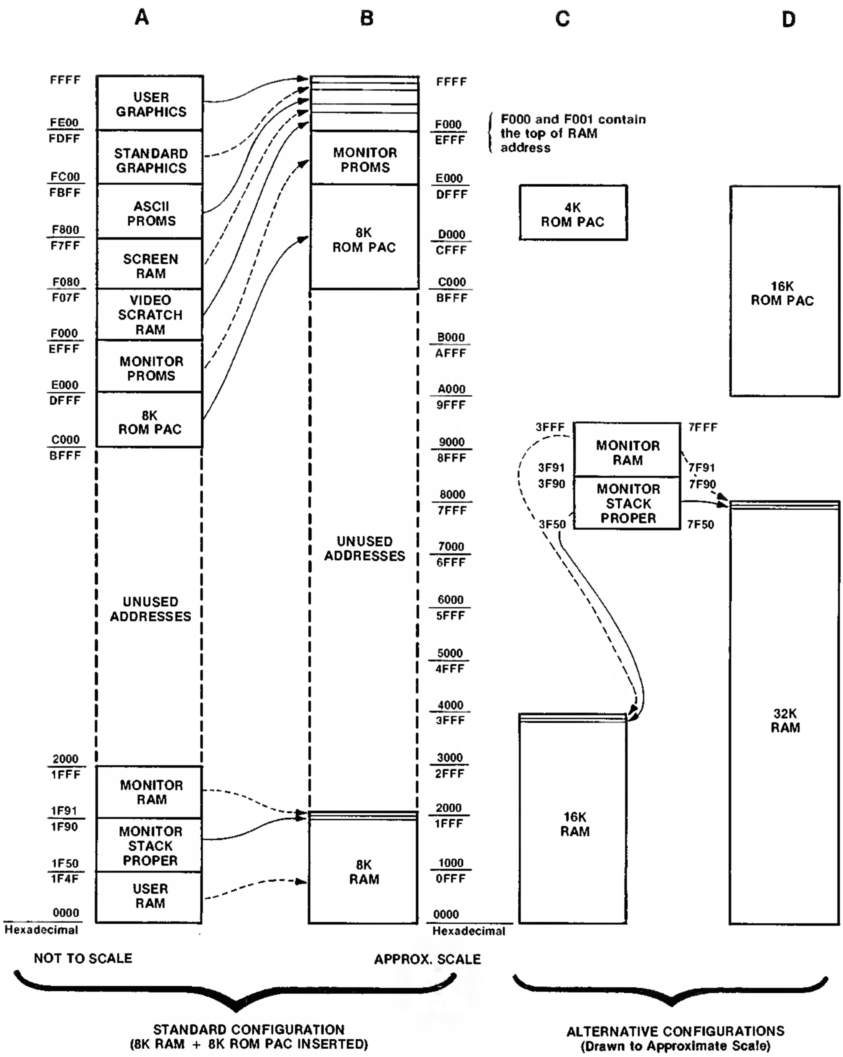


Figure 9. Sorcerer Memory Map

Every character (alphanumeric, graphic, or user-defined) is printed on the screen as an 8 x 8 array of dots. Each dot can be either ON (a white spot on the screen) or OFF (a dark spot). The exact shape of the character depends on which dots are turned on, and which are turned off.

Each row of the character is stored in the Sorcerer's character generator, in a separate memory address, with each dot of the row stored as one bit of that address. A 0 bit means OFF, and a 1 bit means ON.

To write a character onto the Sorcerer's screen, put its ASCII number into the screen RAM. The exact address you use will determine where the character will appear on the screen. The ASCII number directs the Sorcerer to eight successive addresses in the RAM or ROM character generator. These addresses store the eight lines of the character's dot matrix.

Each text line on screen is 64 characters long (40 hexadecimal), and there are 30 lines in all (1E hexadecimal). The first address in screen RAM (F080) is for the upper left corner of the screen. The next address (F081) is for the second place on the first text line, and so on to the end of the line (F0BF). The pattern continues with the first place on the second line (F0C0) and so forth. The last address (F7FF) is for the lower right corner of the screen.

APPENDIX B

Z80 MICROPROCESSOR INSTRUCTION SET

Z80—CPU INSTRUCTIONS SORTED BY OP-CODE

OBJ CODE	SOURCE STATEMENT
00	NOP
018405	LD BC,NN
02	LD (BC),A
03	INC BC
04	INC B
05	DEC B
0620	LD B,N
07	RLCA
08	EX AF,AF'
09	ADD HL,BC
0A	LD A,(BC)
0B	DEC BC
0C	INC C
0D	DEC C
0E20	LD C,N
0F	RRCA
102E	DJNZ DIS
118405	LD DE,NN
12	LD (DE),A
13	INC DE
14	INC D
15	DEC D
1620	LD D,N
17	RLA
182E	JR DIS
19	ADD HL,DE
1A	LD A,(DE)
1B	DEC DE
1C	INC E
1D	DEC E
1E20	LD E,N
1F	RRA
202E	JR NZ,DIS
218405	LD HL,NN
228405	LD (NN),HL
23	INC HL
24	INC H
25	DEC H
2620	LD H,N
27	DAA
282E	JR Z,DIS
29	ADD HL,HL
2A8405	LD HL,(NN)
2B	DEC HL
2C	INC L
2D	DEC L
2E20	LD L,N
2F	CPL
302E	JR NC,DIS
318405	LD SP,NN
328405	LD (NN),A
33	INC SP
34	INC (HL)
35	DEC (HL)
3620	LD (HL),N
37	SCF
382E	JR C,DIS
39	ADD HL,SP
3A8405	LD A,(NN)
3B	DEC SP
3C	INC A
3D	DEC A
3E20	LD A,N
3F	CCF
40	LD B,B
41	LD B,C
42	LD B,D
43	LD B,E
44	LD B,H,NN
45	LD B,L
46	LD B,(HL)
47	LD B,A
48	LD C,B
49	LD C,C
4A	LD C,D
4B	LD C,E
4C	LD C,H
4D	LD C,L
4E	LD C,(HL)
4F	LD C,A
50	LD D,B
51	LD D,C
52	LD D,D
53	LD D,E
54	LD D,H
55	LD D,L
56	LD D,(HL)
57	LD D,A

OBJ CODE	SOURCE STATEMENT
58	LD E,B
59	LD E,C
5A	LD E,D
5B	LD E,E
5C	LD E,H
5D	LD E,L
5E	LD E,(HL)
5F	LD E,A
60	LD H,B
61	LD H,C
62	LD H,D
63	LD H,E
64	LD H,H
65	LD H,L
66	LD H,(HL)
67	LD H,A
68	LD L,B
69	LD L,C
6A	LD L,D
6B	LD L,E
6C	LD L,H
6D	LD L,L
6E	LD L,(HL)
6F	LD L,A
70	LD (HL),B
71	LD (HL),C
72	LD (HL),D
73	LD (HL),E
74	LD (HL),H
75	LD (HL),L
76	HALT
77	LD (HL),A
78	LD A,B
79	LD A,C
7A	LD A,D
7B	LD A,E
7C	LD A,H
7D	LD A,L
7E	LD A,(HL)
7F	LD A,A
80	ADD A,B
81	ADD A,C
82	ADD A,D
83	ADD A,E
84	ADD A,H
85	ADD A,L
86	ADD A,(HL)
87	ADD A,A
88	ADC A,B
89	ADC A,C
8A	ADC A,D
8B	ADC A,E
8C	ADC A,H
8D	ADC A,L
8E	ADC A,(HL)
8F	ADC A,A
90	SUB B
91	SUB C
92	SUB D
93	SUB E
94	SUB H
95	SUB L
96	SUB (HL)
97	SUB A
98	SBC A,B
99	SBC A,C
9A	SBC A,D
9B	SBC A,E
9C	SBC A,H
9D	SBC A,L
9E	SBC A,(HL)
9F	SBC A,A
A0	AND B
A1	AND C
A2	AND D
A3	AND E
A4	AND H
A5	AND L
A6	AND (HL)
A7	AND A
A8	XOR B
A9	XOR C
AA	XOR D
AB	XOR E
AC	XOR H
AD	XOR L
AE	XOR (HL)
AF	XOR A
B0	OR B
B1	OR C
B2	OR D
B3	OR E
B4	OR H
B5	OR L
B6	OR (HL)
B7	OR A

OBJ CODE	SOURCE STATEMENT
B8	CP B
B9	CP C
BA	CP D
BB	CP E
BC	CP H
BD	CP L
BE	CP (HL)
BF	CP A
C0	RET NZ
C1	POP BC
C28405	JP NZ,NN
C38405	JP NN
C48405	CALL NZ,NN
C5	PUSH BC
C620	ADD A,N
C7	RST D
C8	RET Z
C9	RET
CA8405	JP Z,NN
CB00	RLC B
CB01	RLC C
CB02	RLC D
CB03	RLC E
CB04	RLC H
CB05	RLC L
CB06	RLC (HL)
CB07	RLC A
CB08	RRC B
CB09	RRC C
CB0A	RRC D
CB0B	RRC E
CB0C	RRC H
CB0D	RRC L
CB0E	RRC (HL)
CB0F	RRC A
CB10	RL B
CB11	RL C
CB12	RL D
CB13	RL E
CB14	RL H
CB15	RL L
CB16	RL (HL)
CB17	RL A
CB18	RR B
CB19	RR C
CB1A	RR D
CB1B	RR E
CB1C	RR H
CB1D	RR L
CB1E	RR (HL)
CB1F	RR A
CB20	SLA B
CB21	SLA C
CB22	SLA D
CB23	SLA E
CB24	SLA H
CB25	SLA L
CB26	SLA (HL)
CB27	SLA A
CB28	SRA B
CB29	SRA C
CB2A	SRA D
CB2B	SRA E
CB2C	SRA H
CB2D	SRA L
CB2E	SRA (HL)
CB2F	SRA A
CB38	SRL B
CB39	SRL C
CB3A	SRL D
CB3B	SRL E
CB3C	SRL H
CB3D	SRL L
CB3E	SRL (HL)
CB3F	SRL A
CB40	BIT 0,B
CB41	BIT 0,C
CB42	BIT 0,D
CB43	BIT 0,E
CB44	BIT 0,H
CB45	BIT 0,L
CB46	BIT 0,(HL)
CB47	BIT 0,A
CB48	BIT 1,B
CB49	BIT 1,C
CB4A	BIT 1,D
CB4B	BIT 1,E
CB4C	BIT 1,H
CB4D	BIT 1,L
CB4E	BIT 1,(HL)
CB4F	BIT 1,A
CB50	BIT 2,B
CB51	BIT 2,C
CB52	BIT 2,D
CB53	BIT 2,E
CB54	BIT 2,H

OBJ CODE	SOURCE STATEMENT
CB55	BIT 2,L
CB56	BIT 2,(HL)
CB57	BIT 2,A
CB58	BIT 3,B
CB59	BIT 3,C
CB5A	BIT 3,D
CB5B	BIT 3,E
CB5C	BIT 3,H
CB5D	BIT 3,L
CB5E	BIT 3,(HL)
CB5F	BIT 3,A
CB60	BIT 4,B
CB61	BIT 4,C
CB62	BIT 4,D
CB63	BIT 4,E
CB64	BIT 4,H
CB65	BIT 4,L
CB66	BIT 4,(HL)
CB67	BIT 4,A
CB68	BIT 5,B
CB69	BIT 5,C
CB6A	BIT 5,D
CB6B	BIT 5,E
CB6C	BIT 5,H
CB6D	BIT 5,L
CB6E	BIT 5,(HL)
CB6F	BIT 5,A
CB70	BIT 6,B
CB71	BIT 6,C
CB72	BIT 6,D
CB73	BIT 6,E
CB74	BIT 6,H
CB75	BIT 6,L
CB76	BIT 6,(HL)
CB77	BIT 6,A
CB78	BIT 7,B
CB79	BIT 7,C
CB7A	BIT 7,D
CB7B	BIT 7,E
CB7C	BIT 7,H
CB7D	BIT 7,L
CB7E	BIT 7,(HL)
CB7F	BIT 7,A
CB80	RES 0,B
CB81	RES 0,C
CB82	RES 0,D
CB83	RES 0,E
CB84	RES 0,H
CB85	RES 0,L
CB86	RES 0,(HL)
CB87	RES 0,A
CB88	RES 1,B
CB89	RES 1,C
CB8A	RES 1,D
CB8B	RES 1,E
CB8C	RES 1,H
CB8D	RES 1,L
CB8E	RES 1,(HL)
CB8F	RES 1,A
CB90	RES 2,B
CB91	RES 2,C
CB92	RES 2,D
CB93	RES 2,E
CB94	RES 2,H
CB95	RES 2,L
CB96	RES 2,(HL)
CB97	RES 2,A
CB98	RES 3,B
CB99	RES 3,C
CB9A	RES 3,D
CB9B	RES 3,E
CB9C	RES 3,H
CB9D	RES 3,L
CB9E	RES 3,(HL)
CB9F	RES 3,A
CBA0	RES 4,B
CBA1	RES 4,C
CBA2	RES 4,D
CBA3	RES 4,E
CBA4	RES 4,H
CBA5	RES 4,L
CBA6	RES 4,(HL)
CBA7	RES 4,A
CBA8	RES 5,B
CBA9	RES 5,C
CBAA	RES 5,D
CBAB	RES 5,E
CBAC	RES 5,H
CBAD	RES 5,L
CBAE	RES 5,(HL)
CBAF	RES 5,A
CBB0	RES 6,B
CBB1	RES 6,C
CBB2	RES 6,D
CBB3	RES 6,E
CBB4	RES 6,H

OBJ CODE	SOURCE STATEMENT
CBB5	RES 6,L
CBB6	RES 6,(HL)
CBB7	RES 6,A
CBB8	RES 7,B
CBB9	RES 7,C
CBBA	RES 7,D
CBBB	RES 7,E
CBBC	RES 7,H
CBBD	RES 7,L
CBBE	RES 7,(HL)
CBBF	RES 7,A
CBC0	SET 0,B
CBC1	SET 0,C
CBC2	SET 0,D
CBC3	SET 0,E
CBC4	SET 0,H
CBC5	SET 0,L
CBC6	SET 0,(HL)
CBC7	SET 0,A
CBC8	SET 1,B
CBC9	SET 1,C
CBCA	SET 1,D
CBCB	SET 1,E
CBCC	SET 1,H
CBCD	SET 1,L
CBCE	SET 1,(HL)
CBCF	SET 1,A
CBD0	SET 2,B
CBD1	SET 2,C
CBD2	SET 2,D
CBD3	SET 2,E
CBD4	SET 2,H
CBD5	SET 2,L
CBD6	SET 2,(HL)
CBD7	SET 2,A
CBD8	SET 3,B
CBD9	SET 3,C
CBDA	SET 3,D
CBDB	SET 3,E
CBDC	SET 3,H
CBDD	SET 3,L
CBDE	SET 3,(HL)
CBDF	SET 3,A
CBE0	SET 4,B
CBE1	SET 4,C
CBE2	SET 4,D
CBE3	SET 4,E
CBE4	SET 4,H
CBE5	SET 4,L
CBE6	SET 4,(HL)
CBE7	SET 4,A
CBE8	SET 5,B
CBE9	SET 5,C
CBEA	SET 5,D
CBEB	SET 5,E
CBEC	SET 5,H
CBED	SET 5,L
CBEE	SET 5,(HL)
CBEF	SET 5,A
CBF0	SET 6,B
CBF1	SET 6,C
CBF2	SET 6,D
CBF3	SET 6,E
CBF4	SET 6,H
CBF5	SET 6,L
CBF6	SET 6,(HL)
CBF7	SET 6,A
CBF8	SET 7,B
CBF9	SET 7,C
CBFA	SET 7,D
CBFB	SET 7,E
CBFC	SET 7,H
CBFD	SET 7,L
CBFE	SET 7,(HL)
CBFF	SET 7,A
CC8405	CALL Z,NN
CD8405	CALL NN
CE20	ADC A,N
CF	RST 8
D0	RET NC
D1	POP DE
D28405	JP NC,NN
D320	OUT (N),A
D48405	CALL NC,NN
D5	PUSH DE
D620	SUB N
D7	RST 10H
D8	RET C
D9	EXX
DA8405	JP C,NN
DB20	IN A,(N)
DC8405	CALL C,NN
DD09	ADD IX,BC
DD19	ADD IX,DE
DD218405	LD IX,NN
DD228405	LD (NN),IX

OBJ CODE	SOURCE STATEMENT
DD23	INC IX
DD29	ADD IX,IX
DD2A8405	LD IX,(NN)
DD2B	DEC IX
DD3405	INC (IX+d)
DD3505	DEC (IX+d)
DD360520	LD (IX+d),N
DD39	ADD IX,SP
DD4605	LD B,(IX+d)
DD4E05	LD C,(IX+d)
DD5605	LD D,(IX+d)
DD5E05	LD E,(IX+d)
DD6605	LD H,(IX+d)
DD6E05	LD L,(IX+d)
DD7005	LD (IX+d),B
DD7105	LD (IX+d),C
DD7205	LD (IX+d),D
DD7305	LD (IX+d),E
DD7405	LD (IX+d),H
DD7505	LD (IX+d),L
DD7705	LD (IX+d),A
DD7E05	LD A,(IX+d)
DD8605	ADD A,(IX+d)
DD8E05	ADC A,(IX+d)
DD9605	SUB (IX+d)
DD9E05	SBC A,(IX+d)
DDA605	AND (IX+d)
DDAE05	XOR (IX+d)
DDB605	OR (IX+d)
DDBE05	CP (IX+d)
DDE1	POP IX
DDE3	EX (SP),IX
DDE5	PUSH IX
DDE9	JP (IX)
DDF9	LD SP,IX
DDCB0506	RLC (IX+d)
DDCB050E	RRC (IX+d)
DDCB0516	RL (IX+d)
DDCB051E	RR (IX+d)
DDCB0526	SLA (IX+d)
DDCB052E	SRA (IX+d)
DDCB053E	SRL (IX+d)
DDCB0546	BIT 0,(IX+d)
DDCB054E	BIT 1,(IX+d)
DDCB0556	BIT 2,(IX+d)
DDCB055E	BIT 3,(IX+d)
DDCB0566	BIT 4,(IX+d)
DDCB056E	BIT 5,(IX+d)
DDCB0576	BIT 6,(IX+d)
DDCB057E	BIT 7,(IX+d)
DDCB0586	RES 0,(IX+d)
DDCB058E	RES 1,(IX+d)
DDCB0596	RES 2,(IX+d)
DDCB059E	RES 3,(IX+d)
DDCB05A6	RES 4,(IX+d)
DDCB05AE	RES 5,(IX+d)
DDCB05B6	RES 6,(IX+d)
DDCB05BE	RES 7,(IX+d)
DDCB05C6	SET 0,(IX+d)
DDCB05CE	SET 1,(IX+d)
DDCB05D6	SET 2,(IX+d)
DDCB05DE	SET 3,(IX+d)
DDCB05E6	SET 4,(IX+d)
DDCB05EE	SET 5,(IX+d)
DDCB05F6	SET 6,(IX+d)
DDCB05FE	SET 7,(IX+d)
DE20	SBC A,N
DF	RST 18H
E0	RET PO
E1	POP HL
E28405	JP PO,NN
E3	EX (SP),HL
E48405	CALL PO,NN
E5	PUSH HL
E620	AND N
E7	RST 20H
E8	RET PE
E9	JP (HL)
EA8405	JP PE,NN
EB	EX DE,HL
EC8405	CALL PE,NN
ED40	IN B,(C)
ED41	OUT (C),B
ED42	SBC HL,BC
ED438405	LD (NN),BC
ED44	NEG
ED45	RETN
ED46	IM 0
ED47	LD I,A
ED48	IN C,(C)
ED49	OUT (C),C
ED4A	ADC HL,BC
ED4B8405	LD BC,(NN)
ED4D	RETI
ED50	IN D,(C)
ED51	OUT (C),D

OBJ CODE	SOURCE STATEMENT
ED52	SBC HL,DE
ED538405	LD (NN),DE
ED56	IM 1
ED57	LD A,I
ED58	IN E,(C)
ED59	OUT (C),E
ED5A	ADC HL,DE
ED5B8405	LD DE,(NN)
ED5E	IM 2
ED60	IN H,(C)
ED61	OUT (C),H
ED62	SBC HL,HL
ED67	RRD
ED68	IN L,(C)
ED69	OUT (C),L
ED6A	ADC HL,HL
ED6F	RLD
ED72	SBC HL,SP
ED738405	LD (NN),SP
ED78	IN A,(C)
ED79	OUT (C),A
ED7A	ADC HL,SP
ED7B8405	LD SP,(NN)
EDA0	LDI
EDA1	CPI
EDA2	INI
EDA3	OUTI
EDA8	LDD
EDA9	CPD
EDAA	IND
EDAB	OUTD
EDB0	LDIR
EDB1	CPIR
EDB2	INIR
EDB3	OTIR
EDB8	LDDR
EDB9	CPDR
EDBA	INDR
EDBB	OTDR
EE20	XOR N
EF	RST 28H
F0	RET P
F1	POP AF
F28405	JP P,NN
F3	DI
F48405	CALL P,NN
F5	PUSH AF
F620	OR N
F7	RST 30H
F8	RET M
F9	LD SP,HL
FA8405	JP M,NN
FB	EI
FC8405	CALL M,NN
FD09	ADD IY,BC
FD19	ADD IY,DE
FD218405	LD IY,NN
FD228405	LD (NN),IY
FD23	INC IY
FD29	ADD IY,IY
FD2A8405	LD IY,(NN)
FD2B	DEC IY
FD3405	INC (IY+d)
FD3505	DEC (IY+d)
FD360520	LD (IY+d),N
FD39	ADD IY,SP
FD4605	LD B,(IY+d)
FD4E05	LD C,(IY+d)
FD5605	LD D,(IY+d)
FD5E05	LD E,(IY+d)
FD6605	LD H,(IY+d)
FD6E05	LD L,(IY+d)
FD7005	LD (IY+d),B
FD7105	LD (IY+d),C
FD7205	LD (IY+d),D
FD7305	LD (IY+d),E
FD7405	LD (IY+d),H
FD7505	LD (IY+d),L
FD7705	LD (IY+d),A
FD7E05	LD A,(IY+d)
FD8605	ADD A,(IY+d)
FD8E05	ADC A,(IY+d)
FD9605	SUB (IY+d)
FD9E05	SBC A,(IY+d)
FDA605	AND (IY+d)
FDAE05	XOR (IY+d)
FDB605	OR (IY+d)
FDBE05	CP (IY+d)
FDE1	POP IY
FDE3	EX (SP),IY
FDE5	PUSH IY
FDE9	JP (IY)
FDF9	LD SP,IY
FDCB0506	RLC (IY+d)
FDCB050E	RRC (IY+d)
FDCB0516	RL (IY+d)

OBJ CODE	SOURCE STATEMENT
FDCB051E	RR (IY+d)
FDCB0526	SLA (IY+d)
FDCB052E	SRA (IY+d)
FDCB053E	SRL (IY+d)
FDCB0546	BIT 0,(IY+d)
FDCB054E	BIT 1,(IY+d)
FDCB0556	BIT 2,(IY+d)
FDCB055E	BIT 3,(IY+d)
FDCB0566	BIT 4,(IY+d)
FDCB056E	BIT 5,(IY+d)
FDCB0576	BIT 6,(IY+d)
FDCB057E	BIT 7,(IY+d)
FDCB0586	RES 0,(IY+d)
FDCB058E	RES 1,(IY+d)
FDCB0596	RES 2,(IY+d)
FDCB059E	RES 3,(IY+d)
FDCB05A6	RES 4,(IY+d)
FDCB05AE	RES 5,(IY+d)
FDCB05B6	RES 6,(IY+d)
FDCB05BE	RES 7,(IY+d)
FDCB05C6	SET 0,(IY+d)
FDCB05CE	SET 1,(IY+d)
FDCB05D6	SET 2,(IY+d)
FDCB05DE	SET 3,(IY+d)
FDCB05E6	SET 4,(IY+d)
FDCB05EE	SET 5,(IY+d)
FDCB05F6	SET 6,(IY+d)
FDCB05FE	SET 7,(IY+d)
FE20	CP N
FF	RST 38H

Z80–CPU INSTRUCTIONS SORTED BY MNEMONIC

OBJ CODE	SOURCE STATEMENT
8E	ADC A,(HL)
DD8E05	ADC A,(IX+d)
FD8E05	ADC A,(IY+d)
8F	ADC A,A
88	ADC A,B
89	ADC A,C
8A	ADC A,D
8B	ADC A,E
8C	ADC A,H
8D	ADC A,L
CE20	ADC A,N
ED4A	ADC HL,BC
ED5A	ADC HL,DE
ED6A	ADC HL,HL
ED7A	ADC HL,SP
86	ADD A,(HL)
DD8605	ADD A,(IX+d)
FD8605	ADD A,(IY+d)
87	ADD A,A
80	ADD A,B
81	ADD A,C
82	ADD A,D
83	ADD A,E
84	ADD A,H
85	ADD A,L
C620	ADD A,N
09	ADD HL,BC
19	ADD HL,DE
29	ADD HL,HL
39	ADD HL,SP
DD09	ADD IX,BC
DD19	ADD IX,DE
DD29	ADD IX,IX
DD39	ADD IX,SP
FD09	ADD IY,BC
FD19	ADD IY,DE
FD29	ADD IY,IY
FD39	ADD IY,SP
A6	AND (HL)
DDA605	AND (IX+d)
FDA605	AND (IY+d)
A7	AND A
A0	AND B
A1	AND C
A2	AND D
A3	AND E
A4	AND H
A5	AND L
E620	AND N
CB46	BIT 0,(HL)
DDCB0546	BIT 0,(IX+d)
FDCB0546	BIT 0,(IY+d)

OBJ CODE	SOURCE STATEMENT
CB47	BIT 0,A
CB40	BIT 0,B
CB41	BIT 0,C
CB42	BIT 0,D
CB43	BIT 0,E
CB44	BIT 0,H
CB45	BIT 0,L
CB4E	BIT 1,(HL)
DDCB054E	BIT 1,(IX+d)
FDCB054E	BIT 1,(IY+d)
CB4F	BIT 1,A
BC48	BIT 1,B
CB49	BIT 1,C
CB4A	BIT 1,D
CB4B	BIT 1,E
CB4C	BIT 1,H
CB4D	BIT 1,L
CB56	BIT 2,(HL)
DDCB0556	BIT 2,(IX+d)
FDCB0556	BIT 2,(IY+d)
CB57	BIT 2,A
CB50	BIT 2,B
CB51	BIT 2,C
CB52	BIT 2,D
CB53	BIT 2,E
CB54	BIT 2,H
CB55	BIT 2,L
CB5E	BIT 3,(HL)
DDCB055E	BIT 3,(IX+d)
FDCB055E	BIT 3,(IY+d)
CB5F	BIT 3,A
CB58	BIT 3,B
CB59	BIT 3,C
CB5A	BIT 3,D
CB5B	BIT 3,E
CB5C	BIT 3,H
CB5D	BIT 3,L
CB66	BIT 4,(HL)
DDCB0566	BIT 4,(IX+d)
FDCB0566	BIT 4,(IY+d)
CB67	BIT 4,A
CB60	BIT 4,B
CB61	BIT 4,C
CB62	BIT 4,D
CB63	BIT 4,E
CB64	BIT 4,H
CB65	BIT 4,L
CB6E	BIT 5,(HL)
DDCB056E	BIT 5,(IX+d)
FDCB056E	BIT 5,(IY+d)
CB6F	BIT 5,A
CB68	BIT 5,B
CB69	BIT 5,C
CB6A	BIT 5,D
CB6B	BIT 5,E
CB6C	BIT 5,H
CB6D	BIT 5,L
CB76	BIT 6,(HL)
DDCB0576	BIT 6,(IX+d)
FDCB0576	BIT 6,(IY+d)
CB77	BIT 6,A
CB70	BIT 6,B
CB71	BIT 6,C
CB72	BIT 6,D
CB73	BIT 6,E
CB74	BIT 6,H
CB75	BIT 6,L
CB7E	BIT 7,(HL)
DDCB057E	BIT 7,(IX+d)
FDCB057E	BIT 7,(IY+d)
CB7F	BIT 7,A
CB78	BIT 7,B
CB79	BIT 7,C
CB7A	BIT 7,D
CB7B	BIT 7,E
CB7C	BIT 7,H
CB7D	BIT 7,L
DC8405	CALL C,NN
FC8405	CALL M,NN
D48405	CALL NC,NN
CD8405	CALL NN
C48405	CALL NZ,NN
F48405	CALL P,NN
EC8405	CALL PE,NN
E48405	CALL PO,NN
CC8405	CALL Z,NN
3F	CCF
BE	CP (HL)
DDBE05	CP (IX+d)
FDBE05	CP (IY+d)
BF	CP A
B8	CP B
B9	CP C
BA	CP D
BB	CP E
BC	CP H

OBJ CODE	SOURCE STATEMENT
BD	CP L
FE20	CP N
EDA9	CPD
EDB9	CPDR
EDA1	CPI
EDB1	CPIR
2F	CPL
27	DAA
35	DEC (HL)
DD3505	DEC (IX+d)
FD3505	DEC (IY+d)
3D	DEC A
05	DEC B
0B	DEC BC
0D	DEC C
15	DEC D
1B	DEC DE
1D	DEC E
25	DEC H
2B	DEC HL
DD2B	DEC IX
FD2B	DEC IY
2D	DEC L
3B	DEC SP
F3	DI
102E	DJNZ DIS
FB	EI
E3	EX (SP),HL
DDE3	EX (SP),IX
FDE3	EX (SP),IY
08	EX AF,AF'
EB	EX DE,HL
D9	EXX
76	HALT
ED46	IM 0
ED56	IM 1
ED5E	IM 2
ED78	IN A,(C)
DB20	IN A,(N)
ED40	IN B,(C)
ED48	IN C,(C)
ED50	IN D,(C)
ED58	IN E,(C)
ED60	IN H,(C)
ED68	IN L,(C)
34	INC (HL)
DD3405	INC (IX+d)
FD3405	INC (IY+d)
3C	INC A
04	INC B
03	INC BC
0C	INC C
14	INC D
13	INC DE
1C	INC E
24	INC H
23	INC HL
DD23	INC IX
FD23	INC IY
2C	INC L
33	INC SP
EDAA	IND
EDBA	INDR
EDA2	INI
EDB2	INIR
E9	JP (HL)
DDE9	JP (IX)
FDE9	JP (IY)
DA8405	JP C,NN
FA8405	JP M,NN
D28405	JP NC,NN
C38405	JP NN
C28405	JP NZ,NN
F28405	JP P,NN
EA8405	JP PE,NN
E28405	JP PO,NN
CA8405	JP Z,NN
382E	JR C,DIS
182E	JR DIS
302E	JR NC,DIS
202E	JR NZ,DIS
282E	JR Z,DIS
02	LD (BC),A
12	LD (DE),A
77	LD (HL),A
70	LD (HL),B
71	LD (HL),C
72	LD (HL),D
73	LD (HL),E
74	LD (HL),H
75	LD (HL),L
3620	LD (HL),N
DD7705	LD (IX+d),A
DD7005	LD (IX+d),B
DD7105	LD (IX+d),C
DD7205	LD (IX+d),D

OBJ CODE	SOURCE STATEMENT
DD7305	LD (IX+d),E
DD7405	LD (IX+d),H
DD7505	LD (IX+d),L
DD360520	LD (IX+d),N
FD7705	LD (IY+d),A
FD7005	LD (IY+d),B
FD7105	LD (IY+d),C
FD7205	LD (IY+d),D
FD7305	LD (IY+d),E
FD7405	LD (IY+d),H
FD7505	LD (IY+d),L
FD360520	LD (IY+d),N
328405	LD (NN),A
ED438405	LD (NN),BC
ED538405	LD (NN),DE
228405	LD (NN),HL
DD228405	LD (NN),IX
FD228405	LD (NN),IY
ED738405	LD (NN),SP
0A	LD A,(BC)
1A	LD A,(DE)
7E	LD A,(HL)
DD7E05	LD A,(IX+d)
FD7E05	LD A,(IY+d)
3A8405	LD A,(NN)
7F	LD A,A
78	LD A,B
79	LD A,C
7A	LD A,D
7B	LD A,E
7C	LD A,H
ED57	LD A,I
7D	LD A,L
3E20	LD A,N
46	LD B,(HL)
DD4605	LD B,(IX+d)
FD4605	LD B,(IY+d)
47	LD B,A
40	LD B,B
41	LD B,C
42	LD B,D
43	LD B,E
44	LD B,H,NN
45	LD B,L
0620	LD B,N
ED4B8405	LD BC,(NN)
018405	LD BC,NN
4E	LD C,(HL)
DD4E05	LD C,(IX+d)
FD4E05	LD C,(IY+d)
4F	LD C,A
48	LD C,B
49	LD C,C
4A	LD C,D
4B	LD C,E
4C	LD C,H
4D	LD C,L
0E20	LD C,N
56	LD D,(HL)
DD5605	LD D,(IX+d)
FD5605	LD D,(IY+d)
57	LD D,A
50	LD D,B
51	LD D,C
52	LD D,D
53	LD D,E
54	LD D,H
55	LD D,L
1620	LD D,N
ED5B8405	LD DE,(NN)
118405	LD DE,NN
5E	LD E,(HL)
DD5E05	LD E,(IX+d)
FD5E05	LD E,(IY+d)
5F	LD E,A
58	LD E,B
59	LD E,C
5A	LD E,D
5B	LD E,E
5C	LD E,H
5D	LD E,L
1E20	LD E,N
66	LD H,(HL)
DD6605	LD H,(IX+d)
FD6605	LD H,(IY+d)
67	LD H,A
60	LD H,B
61	LD H,C
62	LD H,D
63	LD H,E
64	LD H,H
65	LD H,L
2620	LD H,N
2A8405	LD HL,(NN)
218405	LD HL,NN
ED47	LD I,A

OBJ CODE	SOURCE STATEMENT
DD2A8405	LD IX,(NN)
DD218405	LD IX,NN
FD2A8405	LD IY,(NN)
FD218405	LD IY,NN
6E	LD L,(HL)
DD6E05	LD L,(IX+d)
FD6E05	LD L,(IY+d)
6F	LD L,A
68	LD L,B
69	LD L,C
6A	LD L,D
6B	LD L,E
6C	LD L,H
6D	LD L,L
2E20	LD L,N
ED7B8405	LD SP,(NN)
F9	LD SP,HL
DDF9	LD SP,IX
FDF9	LD SP,IY
318405	LD SP,NN
EDA8	LDD
EDB8	LDDR
FDA0	LDI
FDB0	LDIR
ED44	NEG
00	NOP
B6	OR (HL)
DDB605	OR (IX+d)
EDB605	OR (IY+d)
B7	OR A
B0	OR B
B1	OR C
B2	OR D
B3	OR E
B4	OR H
B5	OR L
E620	OR N
FDBB	OTDR
EDB3	OTIR
FD79	OUT (C),A
ED41	OUT (C),B
ED49	OUT (C),C
ED51	OUT (C),D
FD59	OUT (C),E
ED61	OUT (C),H
ED69	OUT (C),L
D320	OUT (N),A
FDAB	OUTD
EDA3	OUTI
F1	POP AF
C1	POP BC
D1	POP DE
E1	POP HL
DDE1	POP IX
FDF1	POP IY
F5	PUSH AF
C5	PUSH BC
D5	PUSH DE
E5	PUSH HL
DDE5	PUSH IX
FDE5	PUSH IY
CB86	RFS 0,(HL)
DDCB0586	RFS 0,(IX+d)
FDCB0586	RES 0,(IY+d)
CB87	RES 0,A
CB80	RFS 0,B
CB81	RFS 0,C
CB82	RFS 0,D
CB83	RES 0,E
CB84	RES 0,H
CB85	RES 0,L
CB8F	RES 1,(HL)
DDCB058E	RES 1,(IX+d)
FDCB058E	RFS 1,(IY+d)
CB8F	RES 1,A
CB88	RFS 1,B
CB89	RFS 1,C
CB8A	RES 1,D
CB8B	RES 1,E
CB8C	RES 1,H
CB8D	RES 1,L
CB96	RES 2,(HL)
DDCB0596	RES 2,(IX+d)
FDCB0596	RES 2 (IY+d)
CB97	RES 2,A
CB90	RES 2,B
CB91	RES 2,C
CB92	RES 2,D
CB93	RES 2,E
CB94	RFS 2,H
CB95	RES 2,L
CB9E	RES 3,(HL)
DDCB059F	RES 3,(IX+d)
FDCB059E	RES 3,(IY+d)
CB9F	RES 3,A
CB98	RES 3,B

OBJ CODE	SOURCE STATEMENT
CB99	RES 3,C
CB9A	RES 3,D
CB9B	RES 3,E
CB9C	RES 3,H
CB9D	RES 3,L
CBA6	RES 4,(HL)
DDCB05A6	RES 4,(IX+d)
FDCB05A6	RES 4,(IY+d)
CBA7	RES 4,A
CBA0	RES 4,B
CBA1	RES 4,C
CBA2	RES 4,D
CBA3	RES 4,E
CBA4	RES 4,H
CBA5	RES 4,L
CBAE	RES 5,(HL)
DDCB05AE	RES 5,(IX+d)
FDCB05AE	RES 5,(IY+d)
CBAF	RES 5,A
CBA8	RES 5,B
CBA9	RES 5,C
CBAA	RES 5,D
CBAB	RES 5,E
CBAC	RES 5,H
CBAD	RES 5,L
CBB6	RES 6,(HL)
DDCB05B6	RES 6,(IX+d)
FDCB05B6	RES 6,(IY+d)
CBB7	RES 6,A
CBB0	RES 6,B
CBB1	RES 6,C
CBB2	RES 6,D
CBB3	RES 6,E
CBB4	RES 6,H
CBB5	RES 6,L
CBBE	RES 7,(HL)
DDCB05BE	RES 7,(IX+d)
FDCB05BE	RES 7,(IY+d)
CBBF	RFS 7,A
CBB8	RES 7,B
CBB9	RES 7,C
CBBA	RES 7,D
CBBB	RES 7,E
CBBC	RES 7,H
CBBD	RES 7,L
C9	RET
D8	RET C
F8	RET M
D0	RFT NC
C0	RFT NZ
F0	RFT P
E8	RET PE
E0	RET PO
C8	RFT Z
ED4D	RETI
ED45	RETN
CB16	RL (HL)
DDCB0516	RL (IX+d)
FDCB0516	RL (IY+d)
CB17	RL A
CB10	RL B
CB11	RL C
CB12	RL D
CB13	RL E
CB14	RL H
CB15	RL L
17	RLA
CB06	RLC (HL)
DDCB0506	RLC (IX+d)
FDCB0506	RLC (IY+d)
CB07	RLC A
CB00	RLC B
CB01	RLC C
CB02	RLC D
CB03	RLC F
CB04	RLC H
CB05	RLC L
07	RLCA
ED6F	RLD
CB1E	RR (HL)
DDCB051E	RR (IX+d)
FDCB051F	RR (IY+d)
CB1F	RR A
CB18	RR B
CB19	RR C
CB1A	RR D
CB1B	RR E
CB1C	RR H
CB1D	RR L
1F	RRA
CB0E	RRC (HL)
DDCB050F	RRC (IX+d)
FDCB050E	RRC (IY+d)
CB0F	RRC A
CB08	RRC B
CB09	RRC C

OBJ CODE	SOURCE STATEMENT
CB0A	RRC D
CB0B	RRC E
CB0C	RRC H
CB0D	RRC L
0F	RRCA
ED67	RRD
C7	RST 0
D7	RST 10H
DF	RST 18H
E7	RST 20H
EF	RST 28H
F7	RST 30H
FE	RST 38H
CF	RST 8
9E	SBC A,(HL)
DD9E05	SBC A,(IX+d)
FD9E05	SBC A,(IY+d)
9F	SBC A,A
98	SBC A,B
99	SBC A,C
9A	SBC A,D
9B	SBC A,E
9C	SBC A,H
9D	SBC A,L
DE20	SBC A,N
ED42	SBC HL,BC
ED52	SBC HL,DE
ED62	SBC HL,HL
ED72	SBC HL,SP
37	SCF
CBC6	SET 0,(HL)
DDCB05C6	SET 0,(IX+d)
FDCB05C6	SET 0,(IY+d)
CBC7	SET 0,A
CBC0	SET 0,B
CBC1	SET 0,C
CBC2	SET 0,D
CBC3	SET 0,E
CBC4	SET 0,H
CBC5	SET 0,L
CBCF	SFT 1,(HL)
DDCB05CF	SFT 1,(IX+d)
FDCB05CE	SFT 1,(IY+d)
CBCF	SFT 1,A
CBC8	SFT 1,B
CBC9	SET 1,C
CBCA	SET 1,D
CBCB	SET 1,E
CBCC	SET 1,H
CBCD	SET 1,L
CBD6	SET 2,(HL)
DDCB05D6	SET 2,(IX+d)
EDCB05D6	SET 2,(IY+d)
CBD7	SET 2,A
CBD0	SET 2,B
CBD1	SET 2,C
CBD2	SFT 2,D
CBD3	SET 2,F
CBD4	SET 2,H
CBD5	SET 2,L
CBD8	SET 3,B
CBDE	SET 3,(HL)
DDCB05DE	SET 3,(IX+d)
FDCB05DE	SET 3,(IY+d)
CBDF	SET 3,A
CBD9	SET 3,C
CBDA	SET 3,D
CBDB	SET 3,F
CBDC	SET 3,H
CBDD	SET 3,L
CBE6	SET 4,(HL)
DDCB05E6	SET 4,(IX+d)
FDCB05E6	SET 4,(IY+d)
CBE7	SET 4,A
CBE0	SFT 4,B
CBF1	SFT 4,C
CBE2	SET 4,D
CBE3	SET 4,E
CBE4	SET 4,H
CBE5	SET 4,L
CBEF	SFT 5,(HL)
DDCB05EE	SET 5,(IX+d)
FDCB05EF	SET 5,(IY+d)
CBEF	SFT 5,A
CBE8	SET 5,B
CBE9	SET 5,C
CBEA	SET 5,D
CBEB	SET 5,E
CBEC	SET 5,H
CBED	SFT 5,L
CBF6	SET 6,(HL)
DDCB05E6	SET 6,(IX+d)
FDCB05F6	SFT 6,(IY+d)
CBF7	SFT 6,A
CBF0	SET 6,B
CBF1	SET 6,C

OBJ CODE	SOURCE STATEMENT
CBF2	SET 6,D
CBF3	SET 6,E
CBF4	SET 6,H
CBF5	SET 6,L
CBFE	SET 7,(HL)
DDCB05FE	SET 7,(IX+d)
FDCB05FE	SET 7,(IY+d)
CBFF	SET 7,A
CBF8	SET 7,B
CBF9	SFT 7,C
CBFA	SFT 7,D
CBFB	SFT 7,F
CBFC	SET 7,H
CBFD	SET 7,L
CB26	SLA (HL)
DDCB0526	SLA (IX+d)
FDCB0526	SLA (IY+d)
CB27	SLA A
CB20	SLA B
CB21	SLA C
CB22	SLA D
CB23	SLA E
CB24	SLA H
CB25	SLA L
CB2E	SRA (HL)
DDCB052E	SRA (IX+d)
FDCB052E	SRA (IY+d)
CB2F	SRA A
CB28	SRA B
CB29	SRA C
CB2A	SRA D
CB2B	SRA E
CB2C	SRA H
CB2D	SRA L
CB3E	SRL (HL)
DDCB053E	SRL (IX+d)
FDCB053E	SRL (IY+d)
CB3F	SRL A
CB38	SRL B
CB39	SRL C
CB3A	SRL D
CB3B	SRL E
CB3C	SRL H
CB3D	SRL L
96	SUB (HL)
DD9605	SUB (IX+d)
FD9605	SUB (IY+d)
97	SUB A
90	SUB B
91	SUB C
92	SUB D
93	SUB E
94	SUB H
95	SUB L
D620	SUB N
AE	XOR (HL)
DDAE05	XOR (IX+d)
FDAE05	XOR (IY+d)
AF	XOR A
A8	XOR B
A9	XOR C
AA	XDR D
AB	XDR E
AC	XDR H
AD	XDR L
EE20	XDR N

APPENDIX C

HARDWARE EXPANSION
(S-100 Bus)

By now you have become familiar with the Sorcerer and have begun exploring its software. You will also want to investigate the many available hardware options. We will mention a few of these, to give you an idea of what is available.

Exidy S-100 Expansion Unit

To keep your Sorcerer up to date in the ever changing world of microcomputers, Exidy offers an S-100 Expansion Unit. Over 300 manufacturers produce S-100 devices ranging from floppy disk units to music synthesizers to color graphics boards; these are all available to the Sorcerer through the Expansion Unit. Our S-100 unit plugs right into the Sorcerer's 50-pin edge connector. It has its own 12 amp power supply and a mother board with six slots (2 amps per slot). The units can be daisy-chained if more slots are needed.

The S-100 Expansion Unit will support devices which do Direct Memory Access (DMA). S-100 plug-in cards are assigned to blocks of memory addresses or to Z80 I/O ports by means of jumper pads or DIP switches built into the cards. The Expansion Unit bus control circuits allow you to assign an S-100 card to any block of addresses or I/O port not currently used by the Sorcerer.

Exidy I/O Expansion Kit (not needed with the Exidy S-100 Expansion Unit)

The I/O Expansion Kit contains an S-100 interface card and a cable which attaches the card to the Sorcerer's 50-pin edge connector. The interface card has the same bus controllers and buffers as the Exidy S-100 Expansion Unit; it fits into any other S-100 mother board and allows the Sorcerer to control the bus.

Sorcerer Memory Expansion Kit

This kit conains eight 16K RAMs — enough to increase the Sorcerer's internal memory from 8K to 16K, or from 16K to 32K. To increase memory from 8K to 32K, two kits are needed.

Exidy Dual Disk Drive

Our dual disk unit has all the capacity of an 8-inch floppy in a 5¼ inch format. The drive is quad-density with 630,000 bytes on line, easily expandable to over a million bytes. Each drive has 77 tracks of 16 sectors, with 256 bytes per sector, for a total of 315K bytes per drive. Track-to-track positioning time is only 30 milliseconds, and the data transfer rate is 250,000 baud.

The Exidy Dual Disk Drive contains an S-100 controller card, cables, documentation, Disk Operating System software, an Extended Disk BASIC, and an Assembler.

Exidy Line Printer

The Exidy Line Printer prints five-by-seven dot matrix characters with a throughput of up to 90 lines per minute, in formats of 80 to 132 columns; vertical spacing is six lines per inch. Paper tractors on each side of the carriage provide positive feed. The printer takes a cartridge ribbon, so ribbon changes are clean and simple.

Exidy Graphic Printer

The Exidy EX-820 Graphics Printer prints graphics or standard ASCII characters on five-inch wide electrosensitive paper. Three character sizes (80, 40, and 20 columns) and three graphic resolutions (512, 256, and 128 dots) are software selectable. Software commands also shift the printer between the graphic and alphanumeric modes and between standard and reverse printing.

The graphics printer accepts serial or parallel data at 300 or 1200 baud (jumper selectable). Data cables are not included.

APPENDIX D

GLOSSARY OF TERMS

A glossary of computer buzz words will help you understand some of the text you will encounter on your personal computing venture.

ASCII — Acronym for American Standard Code of Information Interchange. Refers to computer industry standardization of computer binary representation of letters, numbers, symbols and certain control characters, so that information can easily be transferred from one computer or peripheral device to another.

Assembler — Large machine-language program that translates human coded instructions into binary information the computer can use. Assembly Language instructions are abbreviated English commands (called mnemonic codes) that can be easily memorized and written by the computer user. Assembly Language programming is fast and concise but somewhat cryptic to read.

Banked Memory — A method of enlarging the usual 64K RAM memory space addressable by 8-bit microprocessors to a much larger range, usually ½ to 1 megabyte. To avoid addressing confusion, boards above 64K are switched on only when needed by software control.

Baud — A term used to define the signalling speed of information in a computer (typically relating to input and output). It is the number of bits of info per second.

Binary — The two-digit (bit) number system based on 0 and 1. Gates are electronic circuits that are either on or off; these two states can represent the binary bits 0 and 1.

Bit — A binary digit (0 or 1).

Bootstrap — A method by which a computer awakens to full awareness once it is turned on. Usually refers to PROM software that initiates this awakening.

Bubble Memory — Computer memory that does not forget what it knows, even when the power is shut off. Bubble memory is mid-way in price and speed between PROM and floppy disk.

Bug — An error in the hardware **or** software of a computer.

Bus — A power or communication line used in common by many parts of a computer. The S-100 Bus is 100 such parallel common communication lines, each of which is capable of carrying one bit or signal through the entire computer.

Byte — A string of eight binary bits.

Call — Any request by a user's program for executive action by the computer Monitor or Operating System.

Central Processing Unit — The nerve center of a computer; the network of electronic circuits that interprets programs and tells a computer how to carry them out.

Chip — Computer jargon for the tiny silicon slices used to make electronic memories and other circuits. A single chip may have as many as 30,000 electronic parts!

Circuit Board — A rigid fiberglass or phenolic card upon which various electronic parts are mounted. Printed or etched copper tracks connect the various parts to one another.

Clock — The master timing circuit for a microprocessor that synchronizes all its operations. Some microprocessors have tuneable clocks so that its operational speed can be increased or decreased by the user for a particular application.

Code — See Program.

Compiler — A language translator program that condenses user-created high-level language programs into something the computer can execute directly in its binary circuitry. See also Interpreter. Some compilers perform syntax abbreviation as an intermediate step and make use of an associated run-time program to perform the final translation of user program to executable binary code.

CPM — Acronym for Control Program Monitor. One of the most popular microcomputer operating systems.

CPU — See Central Processing Unit.

CRT — Acronym for Cathrode Ray Tube; a computer video display terminal (not suitable for ordinary TV reception).

Cursor Control — Ability to move a video display prompt character to any position on the screen, under either keyboard or software command.

Data — Information; often numerical information.

Debug — Correct errors.

Digital Computer — Calculates with discrete numerical information, as opposed to Analog Computer.

Disk — See Floppy Disk.

Disk Drive — Electromechanical mass storage unit.

Diskette — See Floppy Disk.

DMA — Direct Memory Access. A method by which data can be transferred between peripheral devices and internal memory without intervention by the central processor.

DOS — Disk Operating System. A sophisticated monitor capable of accessing, managing and servicing files and data stored on floppy and hard disk subsystems.

Driver — A software driver is a series of instructions the computer follows to reformat data for transfer to and from a particular peripheral device. The electrical and mechanical requirements are different from one kind of device to another and the software drivers are used to standardize the format of data between them and the central processor.

Dumb Terminal — A computer peripheral without any intelligence of its own. The sole function of the terminal is to translate between human visual symbols and keystrokes and the binary language of the computer.

Dynamic Memory — RAM memory that needs to be refreshed every few milliseconds. Most dynamic RAM boards have on-board refresh logic that relieves the central processor of this tedious task.

Firmware — Programming built into the computer to make its operation simpler for the user to understand. Firmware is usually supplied by the manufacturer stored in PROM memory units. See Monitor.

Fixed Point — A convention used to represent non-integer numbers in the computer.

Floating Point — A convention for representing non-integer numbers in the computer using scientific exponential notation.

Floppy Disk — A flexible plastic disk coated with the same magnetic material used to make recording tape. The disk stores computer information on fifty or more tracks around its surface.

Flow Chart — A diagram of geometric shapes connected by arrows that show the progression of a computer program. Flow charts are handy for developing complicated computer programs and illustrating how programs work.

Foreground/Background — A priority system through which current users' needs are serviced first and other tasks are completed more slowly.

Gate — A very simple electronic circuit that is always either **on** or **off.** Clusters of gates can manipulate binary numbers (0 = off; 1 = on). They can also count, do arithmetic, make decisions and store binary numbers. Gates are the basic building blocks of computers.

Handshaking — Interaction between the central processor and peripheral devices. The devices report their status during data transfers so the processor knows when the operation is completed and more data can be transferred.

Hard Copy — The permanent printout of a program or its results produced by a printer connected to a computer.

Hard Sector — Magnetic floppy discs are divided into wedges called sectors which are physically marked by holes punched through the disc to indicate the various sectors. Soft sectoring is a method of determining position of data on the disc by software calculations rather than physical monitoring of the disc.

Hardware — The circuit boards and electronic parts inside a computer.

Hexadecimal — A number system based on powers of 16, and having sixteen digits usually numbered 0 thru 9 and then A thru F. Decimal 178 is represented as B2 in hexadecimal notation. Note that an 8-bit byte can be expressed in two hexadecimal digits.

Input — The means by which data is entered into a computer. Often a keyboard.

Instruction — A statement or command that tells a computer what to do.

Integer — A whole number, positive, negative, or zero.

Interface — A circuit that controls the flow and format of data between the central processor and peripheral devices.

Interpreter — The program stored inside a computer that converts BASIC statements into the computer's machine language.

Interrupt — A signal that interrupts a running program so that some other task can be performed. Sometimes interrupts are given priorities so that the central processor will suspend its current task only if the priority is great enough for immediate execution.

I/O — Input/Ouptut of information in a computer system.

K — Short for **kilo** meaning thousand. Used to designate memory capacity; thus a 4K memory has approximately 4,000 storeage elements.

Keyboard — A typewriter-like panel of switches and keys used to enter programs and data into a computer.

Language — A system of programming instructions easily understood by both the programmer and the computer. A programming language has rules of syntax, which must be followed when writing instructions to the computer. These instructions are translated into machine language instructions, which are sequences of binary numbers and are based on the microprocessor's internal circuitry. Since the circuitry varies from one microprocessor to another, the binary codes are frequently incompatible. The solution to this problem is the ***higher level language***, such as BASIC. With minor variations, a BASIC program written for one computer is understandable to another. This makes it easier for people to share their programing efforts with others.

Line Printer — See Hard Copy.

Linking Loader — An executive program which connects different program segments so they may be run in the computer as one unit. A useful piece of software that makes subtasks easily available to a main task.

Macro-Assembler — An Assembler that allows the user to create and define new computer instructions (called macros).

Mainframe — The box with power supply, motherboard and (optional) front panel, into which various printed circuit boards may be plugged.

Megabyte — A million keystrokes; a million characters.

Memory — Any of the many devices (ROMs, RAMs, floppy disks, magnetic tapes, etc.) that store computer programs and data.

MHz — Megahertz. One Hertz equals one cycle per second. Notation for the frequency or clock speed of various integrated circuits, e.g., the 8080A operates at 2 MHz and the Z-80 at 4 MHz.

Microprocessor — The central processing unit of a computer assembled on a single silicon chip.

Microcomputer — A computer made by combining a microprocessor with some memory. Microcomputers are small in size, not performance.

Modem — Electronic equipment hard-wired into a telephone line to facilitate connection and disconnection between a computer and remote peripherals.

Monitor — A small package of software usually stored on PROMs that gives the computer a fundamental interactive intelligence. The Monitor usually contains software routines and I/O drivers needed by the user to operate the system. The Monitor is a kind of executive secretary for the user. It tells the computer how and where to acquire the programs and data, where to store them, and how to run them.

Motherboard — The central communications bus line. The spinal cord of a microcomputer.

NRZ — Non Return to Zero, one of several methods for coding digital information on magnetic tape.

Operating System — A sophisticated monitor often found with floppy disc systems.

Output — The means by which data leaves a computer. Often a television monitor or printer.

Paper Tape — A narrow ribbon of paper which contains computer data in the form of punched holes. A hole indicates the bit 1; no hole indicates the bit 0. Paper tape is sometimes used to enter programs into a computer.

Parallel — A type of interface in which all bits of data in a given byte are transferred simultaneously, using a separate data line for each bit.

Parity Bits — Used to ensure integrity of data transmitted along communications lines.

Peripheral — An accessory which can be added to a computer to increase its capability and usefulness (a floppy disk, paper tape unit, etc.)

Personal Computer — An economical microcomputer designed for use by small businesses, schools and computer hobbyists.

Port — The physical communication line between the central processor and a peripheral. Each port has a numerical address that the processor uses in communicating through it. 8080/Z-80 microprocessors can address up to 256 input and 256 output ports.

Printed Circuit — Electrical connections of solder traces between components on an epoxy board material.

Printer — A computer output mechanism that delivers hard copy data.

Program — The list of instructions or statements that tells a computer what to do.

Program Cartridge — Read Only Memory on a printed circuit board enclosed in a cartridge case.

Programmer — A person who writes programs. Usually a professional.

PROM — Programmable Read Only Memory. This is computer memory which does not forget what it knows, even when the power is shut off. Some kinds of PROM can be erased and reused: EPROMs, or Erasable PROMs. PROMs are a convenient way for the user to design his own operating system software and other tailor-made monitor routines.

Prompt — A character displayed by a computer as a signal that it expects a response.

RAM — (**R**andom **A**ccess **M**emory) A temporary memory, i.e., one in which data can be stored so long as power is applied. RAMs store the data that's typed into the keyboard of a microcomputer.

Real-Time Clock — A piece of hardware which interrupts the processor at fixed time intervals to synchronize the operations of the computer with events occurring in the outside world, often involving man/computer interaction.

Reentrant Code — Assembler-generated machine language programs that may be shared simultaneously by any number of users.

Refresh — A signal sent to dynamic RAM every few milliseconds to help it remember data.

Relocatable Code — Assembler-generated machine language programs that may be placed anywhere in available computer memory for execution. Relocatable code makes life easier for the small system user who does not possess a full complement of memory boards.

ROM — (**R**ead **O**nly **M**emory) A permanent memory, i.e., one in which data is stored permanently whether or not electrical power is applied.

ROM PAC™ — Read Only Memory programmed and packaged on a printed circuit card inside a plastic cartridge.

RS232 — An industry-wide standard protocol for serial communication between computers and peripheral devices.

Serial — A type of interface in which all the bits of data in a given byte are lined up sequentially (usually with start, stop and parity-error checking bits at the head and tail of the byte) for transfer along one data line.

Smart Terminal — A computer peripheral capable of computing functions on its own. Smart terminals can usually be switched on command to dumb terminal mode for conversation with larger computers.

Software — Computer programs written on paper or stored on magnetic tape or a floppy disk.

Standard BASIC — A set of easy English word instructions used to program a microcomputer.

Statement — A single line of a computer program containing a single instruction like PRINT, LET, RUN, etc.

Static Memory — RAM memory that holds information with high reliability as long as power is applied. More expensive than dynamic RAM.

String — A group of data elements (usually ASCII characters) stored in sequential memory locations and treated as one unit for I/O operations, text editing and other program manipulations.

System — A complete, integrated and functional computer made up of various hardware components linked harmoniously together and unified by software-programmed intelligence.

Telecommunications — Communication between computers and peripheral devices over telephone lines.

Terminal — An input device such as a keyboard; an output device such as a printer or TV monitor; or both.

Timesharing — A computer system that seems to be performing multiple tasks for a number of users simultaneously. In actuality, the processor is working for only one user at any particular moment, but has been programmed to remember what each user was doing last.

Turn-key — A computer system ready to perform all tasks the moment you turn it on. Business and accounting software is frequently supplied in ready-to-run form on such a system.

User's Group — An informal or formal association of persons who own or operate similar or identical computing equipment. User's groups are usually formed to exchange programs and other helpful information.

Vectored Interrupt — A procedure by which interrupts force the central processor to transfer program control to a particular pre-stored routine that handles and processes the interrupt.

Virtual Memory — A method by which maxi- and minicomputers appear to the user to have unlimited RAM memory resources. Hard disk storage space is used to extend the mainframe RAM through sophisticated operating system program segmentation techniques.

APPENDIX E
ASCII CHARACTER CODE

ASCII stands for American Standard Code for Information Exchange.

Decimal and Hexadecimal Designations of the Standard Characters

CTRL	DEC.	HEX.	CHAR.	DEC.	HEX.	CHAR.	DEC.	HEX.	CHAR.
@	000	00	NUL	031	1F	US	062	3E	>
A	001 *	01	SOH	032	20	SPACE	063	3F	?
B	002	02	STX	033	21	!	064	40	@
C	003 Quick Clr	03	ETX	034	22	"	065	41	A
D	004	04	EOT	035	23	#	066	42	B
E	005	05	ENQ	036	24	$	067	43	C
F	006	06	ACK	037	25	%	068	44	D
G	007	07	BEL	038	26	&	069	45	E
H	008 *	08	BS BS	039	27	'	070	46	F
I	009	09	HT	040	28	(	071	47	G
J	010 *	0A	LF LF	041	29	)	072	48	H
K	011	0B	VT	042	2A	*	073	49	I
L	012 *	0C	FF Home & Clear	043	2B	+	074	4A	J
M	013 *	0D	CR	044	2C	,	075	4B	K
N	014	0E	SO	045	2D	–	076	4C	L
O	015	0F	SI	046	2E	.	077	4D	M
P	016	10	DLE	047	2F	/	078	4E	N
Q	017 *	11	DC1 Home	048	30	0	079	4F	O
R	018	12	DC2	049	31	1	080	50	P
S	019 *	13	DC3	050	32	2	081	51	Q
T	020	14	DC4	051	33	3	082	52	R
U	021	15	NAK	052	34	4	083	53	S
V	022	16	SYN	053	35	5	084	54	T
W	023 *	17	ETB	054	36	6	085	55	U
X	024	18	CAN	055	37	7	086	56	V
Y	025	19	EM	056	38	8	087	57	W
Z	026 *	1A	SUB LF	057	39	9	088	58	X
[	027 Shift Clr	1B ESC R/S	ESC	058	3A	:	089	59	Y
\	028	1C	FS	059	3B	;	090	5A	Z
]	029	1D	GS	060	3C	<	091	5B	[
↑	030	1E	RS	061	3D	=	092	5C	\

* May be off by one

DEC.	HEX.	CHAR.	DEC.	HEX.	CHAR.	DEC.	HEX.	CHAR.
093	5D	]	105	69	i	117	75	u
094	5E	^	106	6A	j	118	76	v
095	5F	<	107	6B	k	119	77	w
096	60	'	108	6C	l	120	78	x
097	61	a	109	6D	m	121	79	y
098	62	b	110	6E	n	122	7A	z
099	63	c	111	6F	o	123	7B	{
100	64	d	112	70	p	124	7C	¦
101	65	e	113	71	q	125	7D	}
102	66	f	114	72	r	126	7E	~
103	67	g	115	73	s	127	7F	DELETE
104	68	h	116	74	t			(Rubout)

Standard Abbreviations for ASCII characters 0 through 31 (00 through 1F Hex.)

ACK = Acknowledge
BELL = Bell
BS = Backspace
CAN = Cancel
CR = Carriage Return
DC1 = Direct Control 1 , HOME
DC2 = Direct Control 2
DC3 = Direct Control 3 →
DC4 = Direct Control 4
DLE = Data Link Escape
EM = End of Medium
ENQ = Enquiry
EOT = End Of Transmission
ESC = Escape
ETB = End Transmission Block ↑
ETX = End Text

FF = Form Feed
FS = Form Separator
GS = Group Separator
HT = Horizontal Tab
LF = Line Feed
NAK = Negative Acknowledge
NUL = Null
RS = Record Separator
SI = Shift In *
SO = Shift Out *
SOH = Start of Heading ←
STX = Start Text
SUB = Substitute , LF
SYN = Synchronous Idle
US = Unit Separator
VT = Vertical Tab

* To get out a) Depress Run/Stop > 3 sec.
b) Depress C, O <CR>
c) Depress : <CR>
d) Depress R/S, CLEAR <CR>
e) Repeat CTRL O

LIMITED WARRANTY

EXIDY warrants to the registered owner for a period of 90 days from the date of purchase that the computer components described herein shall be free from defects in material and workmanship under normal use and service. This warranty shall be void if the computer enclosure is opened or if the unit is altered or returned as outlined in Customer Service Information. (Refer to A GUIDED TOUR OF PERSONAL COMPUTING, p. 5.) Customer's sole and exclusive remedy in the event of defect is expressly limited to the correction of the defect by adjustment, repair or replacement at EXIDY's election and there shall be no obligation to replace or repair items which by their nature are expendable. No representation or other affirmation of fact, including but not limited to statements regarding capacity, suitability for use, or performance of the equipment, shall be deemed to be a warranty or representation by EXIDY, for any purpose, nor give rise to any liability or obligation of EXIDY what-so-ever.

EXCEPT AS SPECIFICALLY PROVIDED IN THIS AGREEMENT, THERE ARE NO OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OR MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE AND IN NO EVENT SHALL EXIDY BE LIABLE FOR LOSS OF PROFITS OR BENEFITS, INDIRECT, SPECIAL, CONSEQUENTIAL OR OTHER SIMILAR DAMAGES ARISING OUT OF ANY BREACH OF THIS WARANTY OR OTHERWISE.

NOTICE

ALL EXIDY SOFTWARE IS DISTRIBUTED ON AN "AS IS" BASIS WITHOUT WARRANTY

EXIDY shall have no liability or responsibility to customers or any other person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by computer equipment or software sold or distributed by EXIDY, including but not limited to any interruption of services, loss of business or anticipatory profits or consequential damages resulting from the use or operation of such computer or software.

$9.95

inc.

REORDER DP 5001